1990年10月1日発行（毎月1回1日発行）第9巻10号通巻102号 昭和58年11月2日第三種郵便物認可

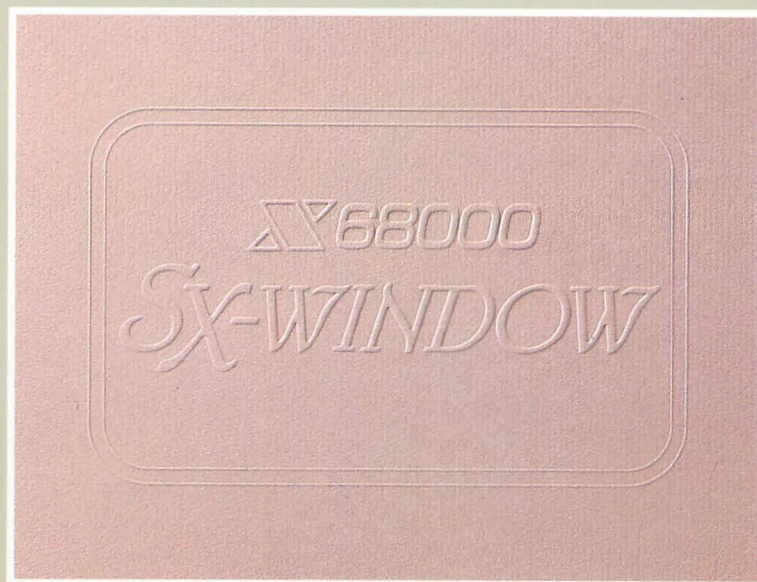
PERSONAL COMPUTER MAGAZINE for MZ, X1, and X68000

特集 電子音楽術入門

新連載 中森章の「ようこそここへC言語」
ASK68K用辞書ユーティリティ後編
詳報 C compiler PRO-68K ver. 2.0

10
1990

オー／エックス
定価560円

ひらかれた知性。

ザ・ワークステーション。80Mバイト(SCSI仕様)ハードディスク、SCSIインターフェイスを標準装備。

SUPER HD

本体＋キーボード＋マウス・トラックボール

CZ-623C-TN(チタンブラック) 標準価格498,000円(税別)

アートの系譜。

EXPERT II

本体＋キーボード＋マウス・トラックボール

CZ-603C-BK(ブラック)・-GY(グレー) 標準価格338,000円(税別)/HDタイプ CZ-613C-BK(ブラック) 標準価格448,000円(税別)

ニュースタンダード。

PRO II

本体＋キーボード＋マウス

CZ-653C-BK(ブラック)・-GY(グレー) 標準価格285,000円(税別)

HDタイプ CZ-663C-BK(ブラック)・-GY(グレー) 標準価格395,000円(税別)

次代のユーザーインターフェイスを象徴する"SX-WINDOW※"搭載。

今回のX68000ニューシリーズのデビューに関して、ハードウェア以上にウィンドウ環境の提供に耳目が集中したことは、昨今のビジュアルユーザーインターフェイス事情をふまえれば、当然のことと言えるでしょう。マルチウィンドウを駆使してX68000をコントロールする、待ち望まれていた環境がこのSX-WINDOWによって実現されるのです。何の予備知識もなしにこのウィンドウに接した方は、一見して従来のビジュアルシェルのバージョンアップと思われるかもしれませんが、本質的には全く異質のものと言えます。ひとつのウィンドウである仕事をさせながら、別のウィンドウで違う仕事にとりかかる。ひとことで言えばアプリケーションを実行させる環境としてのウィンドウであるということ。これまでのビジュアルシェルではできなかったシーンを生み出しています。複数のアプリケーションを同じ操作のもとで走らせたり、アプリケーション相互でデータのやりとりが可能になるわけです。そして、次代のインテリジェンスを鮮やかに象徴する4階調のハイセンスな画面処理——。SX-WINDOWをターゲットとしたアプリケーション開発もすでに推進されており、これからの展望という点からも大いに期待されるところです。また、このSX-WINDOWはディスクによって供給され、BIOSの高速化(平均2倍)も含めてOSであるHuman68kの機能を拡張。ニューシリーズのみならず、すべてのX68000でこの新しい環境が享受できます。

※SX-WINDOWの起動には、メインメモリ2MBが必要です。CZ-600C/601C/611C/652C/653C/662C/663CでSX-WINDOWをご使用の際には、あらかじめ別売の1MB増設RAMボードを増設してください。

SUPER・EXPERT・PRO

—充実のディスプレイラインアップ—

15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-602D-BK(ブラック)・-GY(グレー)※	標準価格 99,800円(チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.39mm)	CZ-605D-BK(ブラック)・-GY(グレー)	標準価格115,000円(スピーカー2個/チルトスタンド同梱・税別)
15型カラーディスプレイテレビ(ドットピッチ0.31mm)	CZ-613D-TN(チタンブラック)・-BK(ブラック)・-GY(グレー)	標準価格135,000円(スピーカー2個/チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-603D-BK(ブラック)・-GY(グレー)	標準価格 84,800円(チルトスタンド同梱・税別)
14型カラーディスプレイ(ドットピッチ0.31mm)	CZ-604D-BK(ブラック)・-GY(グレー)	標準価格 94,800円(スピーカー2個/チルトスタンド同梱・税別)
21型カラーディスプレイ(ドットピッチ0.52mm)	CU-21HD-BK(ブラック)	標準価格148,000円(スピーカー2個同梱・税別)

※印の商品は在庫僅少です。

シャープX68000 パソコン教室開催中
●会場:市ヶ谷教室 シャープ東京支社ビル●コース:入門コース・表集計コース・音楽コース・絵画コース●申込受付電話番号:(03)260-8365●受講料:2,000円(税別)

EXEリーダーズグッズ プレゼント実施中
●いま、EXE会員よりご紹介のお客様がEXEショップでX68000シリーズを購入されますと、EXE会員にEXEリーダーズグッズをプレゼントします。詳しくはEXEショップにお問い合わせください。●また、X68000シリーズをご購入のお客様は、ぜひEXEクラブにご入会ください。

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)
シャープ株式会社

C-TRACE68 TP

ひとり占いTEN

それ行け！ ロケット

ルーンワース

闇の血族

C compiler PRO-68K ver.2.0

Oh!X

CONT

〈スタッフ〉
●編集長／前田 徹 ●副編集長／植木章夫 ●編集／岡崎栄子 浅井研二 ●協力／有田隆也 中森 章 後藤貴行 林 一樹 荻窪 圭 岡本浩一郎 毛内俊行 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 丹 明彦 三沢和彦 長沢淳博 宮島 靖 金子俊一 浦川博之 山田純二 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／グループごじら

表紙絵：須藤　牧人

1990 OCT. 10

E N T S

■広告目次

X68000をサポート。

シャープペリフェラルファミリー

X68000

ボード

拡張メモリ

1MB増設RAMボード
(CZ-600C専用)
CZ-6BE1
標準価格 35,000円(税別)

1MB増設RAMボード
(CZ-601C/611C/652C/653C/662C/663C用)
CZ-6BE1B
標準価格 28,000円(税別)

2MB増設RAMボード※5
CZ-6BE2
標準価格 79,800円(税別)

4MB増設RAMボード※5
CZ-6BE4
標準価格 138,000円(税別)

インターフェイス

ユニバーサルI/Oボード
CZ-6BU1
標準価格 39,800円(税別)

GP-IBボード
CZ-6BG1
標準価格 59,800円(税別)

増設用RS-232Cボード
(2チャンネル)
CZ-6BF1
標準価格 49,800円(税別)

数値演算プロセッサ

数値演算プロセッサボード
CZ-6BP1
標準価格 79,800円(税別)

FAX

FAXボード
CZ-6BC1
標準価格 79,800円(税別)

MIDI

MIDIボード
CZ-6BM1
標準価格 26,800円(税別)

ネットワーク

モデム

モデムユニット※6
CZ-8TM2
標準価格 49,800円(税別)
(RS-232Cケーブル同梱)

RS-232Cケーブル

RS-232Cケーブル
(平行接続型)
CZ-8LM1
標準価格 7,200円(税別)

RS-232Cケーブル
(クロス接続型)
CZ-8LM2
標準価格 7,200円(税別)

LANボード

NEW

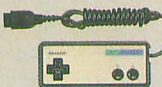
LANボード
CZ-6BL1
標準価格268,000円(税別)
(イーサネット用)
CZ-6BL2
標準価格298,000円(税別)
(イーサネット/チーパネット両用)
※電源ユニット・ソフトウェア
(ネットワークドライバVer1.0)同梱

入力

インテリジェントコントローラ
CZ-8NJ2
標準価格 23,800円(税別)

マウス・トラックボール
CZ-8NM3
標準価格9,800円(税別)

トラックボール
CZ-8NT1
標準価格 13,800円(税別)

マウス
CZ-8NM2A
標準価格 6,800円(税別)

ジョイカード
CZ-8NJ1
標準価格 1,700円(税別)

その他

拡張スロット

拡張 I/Oボックス(4スロット)
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6EB1-BK
CZ-6EB1
標準価格 88,000円(税別)

スピーカー

アンプ内蔵
スピーカーシステム(2本1組)
AN-S100
標準価格 36,600円(税別)

システムラック

システムラック
(CZ-600C/601C/602C/603C/611C/612C/613C/623C用)
CZ-6SD1
標準価格 44,800円(税別)

※4 別売の信号ケーブルIO-73CX標準価格5,500円(税別)で接続して下さい。
※5 ご使用に際しては、あらかじめ別売の1MB増設RAMボードCZ-6BE1 標準価格35,000円(税別・CZ-600C用)、CZ-6BE1B 標準価格28,000円(税別・CZ-601C、CZ-611C、652C、653C、662C、663C用)を増設してください。
※6 モデムユニットCZ-8TM2に同梱のソフトはX1/X1ターボシリーズ用です。

●ミニフロッピーディスクユニット(2D)	★CZ-502F	99,800円
●ミニフロッピーディスクユニット(2D・1ドライブ)	CZ-503F	49,800円
●増設用ミニフロッピーディスクドライブ(2D)※4	CZ-53F-BK	19,800円

拡張ボード・その他

●モデムユニット(300/1200ボー)	CZ-8TM2	49,800円
●320KB外部メモリ	CZ-8BE2	29,800円
●RS-232C・マウスボード※5	CZ-8BM2	19,800円
●フロッピーディスクインターフェイス※6	CZ-8BF1	14,800円

●JIS第1水準漢字ROM※7	CZ-8BK2	19,800円
●RS-232C用ケーブル(平行接続型)	CZ-8LM1	7,200円
●RS-232C用ケーブル(クロス接続型)	CZ-8LM2	7,200円
●拡張 I/Oボックス	CZ-8EB3	33,800円
●RFコンバータ※8	AN-58C	2,980円
●インテリジェントコントローラ	CZ-8NJ2	23,800円
●マウス・トラックボール	CZ-8NM3	9,800円
●マウス	CZ-8NM2A	6,800円
●トラックボール	CZ-8NT1	13,800円

●ジョイカード	CZ-8NJ1	1,700円
●チルトスタンド	CZ-6ST1-E・-B	5,800円
●高性能CRTフィルター ※9	BF-68PRO	19,800円
●スキャナ用パラレルボード※10	CZ-8BN1	27,800円

●品番中の-表示は、B〈ブラック〉・E〈オフィスグレー〉を示します。※1 X1ターボZシリーズ用 ※2 CZ-862Cには接続できません ※3 X1ターボシリーズ用 ※4 CZ-830C用 ※5 X1シリーズ用 ※6 CZ-850CでCZ-520Fを使用する場合に必要 ※7 CZ-800C、801C、802C、803C、811C、820C用 ※8 CZ-820C、822C、830C用 ※9 14/15型用 ※10 CZ-8NS1用 ●接続等の説明につきましては、周辺機器総合カタログをご参照ください。

★印の商品は在庫僅少です。

子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

SHARP

"アート"と呼べる高水準のソフトウェアが

次代のインテリジェンス、ウィンドウ環境をあなたのX68000で。

ユーザー本位の操作環境を提供するフル画面マルチウィンドウタイプの美しいデスクトップ(テキスト面/単色4階調＋カラー4色、グラフィック面/カラー65,536色中16色)、イベント・ドリブン型マルチタスク処理により複数の作業を同時に処理できる疑似マルチタスクや入出力装置の設定が簡単に行える多機能コントロールパネルを搭載した本格ウィンドウシステムです。IOCSコールを利用したソフトの処理速度を高速化するIOCS.Xを付属。

SX-WINDOW ver1.0

CZ-259SS 10万台達成ご愛用感謝価格6,800円(税別)

高速通信をサポート。これからの、そしてさまざまな通信環境に対応する高機能コミュニケーションソフト。

Communication PRO-68Kのバージョンアップ版です。300BPSから19,200BPSまでの通信速度に対応し、パソコン同士の接続や各種データベースの漢字端末に、またホストコンピュータとのデータ通信に利用できます。さらにMNPモデムへの対応で、ハードフロー制御(CTS/RTS)をサポート。その他、高速逆スクロール機能、オートログイン/オートパイロットが可能な自動実行機能、コンカレント機能も装備。また、バイナリファイルを転送するプロトコルとしてX modem(128/SUM、128/CRC、1K)、Ymodem(G、BATCH、G-BATCH)、TransIt2(TEXT、BINARY)プロトコルもサポート。

※Communication PRO-68K(CZ-223CS)を既にお持ちの方は、アンケートカードをもとに有償バージョンアップを行います。

CZ-257CS

標準価格 19,800円(税別)

Communication PRO-68K ver2.0

ソースコードデバッガをはじめ、各種開発ツールを強化。バージョンアップされたCコンパイラ。

Cのソースレベルでデバッグできる「ソースコードデバッガ」を搭載したほか、各種開発ツールを強化した総合開発ツールです。また、ライブラリはHuman 68k ver 2.0の拡張DOSコールもサポートしているなど、よりX68000のハードウェアを活かせる豊富なライブラリ(800種以上)となっています。C言語の標準であるANSI規格準拠をさらに強化。「プログラム保守ユーティリティ(MAKE)」や「ライブラリアン」など各種ツールを追加しました。「BASIC-Cコンバータ」、「アセンブラ」、「リンカ」、「デバッガ」、「ソースコードデバッガ」、「アーカイバ」、「コンバータ」、などのツールが装備されています。

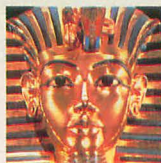
※C Compiler PRO-68K(CZ-211LS)を既にお持ちの方は、登録カードをもとに有償バージョンアップを行います。

CZ-245LS

標準価格 44,800円(税別)

C COMPILER PRO-68K ver2.0

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部液晶映像システム事業部第2商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券 CZソフト Oh!X 10係

X68000をサポート。

シャープオリジナルソフトウェア
X68000

ビジネスツール

Hyperword
■CZ-251BS 標準価格39,800円(税別)

X68000の優れたグラフィック環境を活用し効率的に文書を作成するためのインテリジェントワープロです。アイデアプロセッサ機能、ハイパーテキスト機能などをサポート。データの整理やプレゼンテーションツールなど幅広い用途に利用できます。

CYBERNOTE PRO-68K
■CZ-243BS 標準価格19,800円(税別)

プライベートなデータやビジネスデータを簡単な操作で管理・運営できるパーソナルデータベースです。リフィル、タックシール、ハガキなどへの印字もOK。シャープ電子手帳とのデータ交換可能(別売の通信ケーブルCE-200Lが必要)。

Stationery PRO-68K
■CZ-240BS 標準価格14,800円(税別)

他のソフトを起動する前に、このStationery PRO-68Kを一度起動するだけで、他のソフトを実行中にも「スケジュール」「住所録」など多彩な機能をワンタッチで使用できます。シャープ電子手帳とのデータ送受信も実現。(別売の通信ケーブルCE-200Lが必要)。

TOP給与計算エキスパート
■CZ-228BS標準価格200,000円(税別)

給与計算から明細発行までを、リアルイメージ入力により自動的に、素早く処理することができます。

CARD PRO-68K
■CZ-226BS 標準価格29,800円(税別)

自由なレイアウト画面で入力できるワープロ機能を装備したカード型リレーショナルデータベース。

DATA PRO-68K
■CZ-220BS 標準価格58,000円(税別)

入力の手間を軽減するヒストリー機能を装備した、コマンド型リレーショナルデータベースです。

TOP財務会計
■CZ-227BS標準価格200,000円(税別)

会計エキスパートシステムとデータベースを搭載し、機能と操作性を両立させた財務会計ソフト。

CARD PRO-68K用システム手帳リフィル集
■CZ-241BS 標準価格9,800円(税別)

CARD PRO-68K用活用フォーム集
■CZ-242BS 標準価格9,800円(税別)

BUSINESS PRO-68K
■CZ-212BS 標準価格68,000円(税別)

スプレッドシート(表計算)、データベース、グラフ作成機能を一体化させた統合ビジネスツールです。

アートツール

NEW PrintShop PRO-68K
■CZ-221HS 標準価格19,800円(税別)

オリジナリティあふれるはがき等、簡単に作成、印刷できるホームプロダクティビリティツール。ほとんどの処理をアイコンで表示しマウスで選ぶフレンドリーオペレーション。

グラフィックライブラリ VOL.1
■CZ-235GS 標準価格8,800円(税別)

暑中見舞用を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

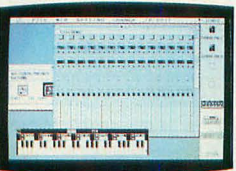
グラフィックライブラリ VOL.2
■CZ-236GS 標準価格8,800円(税別)

年賀状を中心としたNEW Print Shop PRO-68K用グラフィックデータ集。

サウンドツール

Musicstudio PRO-68K ver.1.1
■CZ-252MS 標準価格28,800円(税別)

24トラック対応MIDIマルチレコーディングソフトMusicstudio PRO-68Kがバージョンアップしました。従来の機能に加え、小節間のコピー及びデリートや、MIDIインプットモニターなど、数々の機能を追加・改良。さらに使いやすくなりました。

※MIDIボード(CZ-6BM1)が必要です。

MUSIC PRO-68K〔MIDI〕
■CZ-247MS 標準価格28,800円(税別)

MIDI対応自動伴奏機能をサポート、簡単な楽譜入力でMIDI演奏が楽しめます。

※MIDIボード(CZ-6BM1)が必要です。

ソングライブラリ〈101曲集〉
■CZ-248MS 標準価格8,800円(税別)

鑑賞用と音楽データ加工作成用からなるライブラリです。

Sampling PRO-68K
■CZ-215MS 標準価格17,800円(税別)

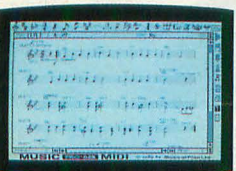
AD PCM機能を活かす高機能サンプリングエディタ。多彩なEDITORを装備、サンプリング音のデータはBASICでも活用できます。

SOUND PRO-68K
■CZ-214MS 標準価格15,800円(税別)

スタジオのコンソールパネルを操作する感覚でFM音源による音創りが楽しめるサウンドエディタ。

MUSIC PRO-68K
■CZ-213MS 標準価格18,800円(税別)

最大8パートのスコア(総譜)が書け、内蔵のFM音源で演奏できる楽譜ワープロ＆演奏用ツール。

ゲーム

シューティングゲーム
〈ツインビー〉
■CZ-217AS
標準価格7,800円(税別)
©KONAMI 1988

シューティングゲーム
〈沙羅曼蛇〉
■CZ-218AS
標準価格8,800円(税別)
©KONAMI 1989

ブロックゲーム
〈アルカノイド〉
■CZ-222AS
標準価格7,800円(税別)
©TAITO CORP. 1987

ドライブゲーム
〈フルスロットル〉
■CZ-231AS
標準価格8,800円(税別)
©TAITO CORP. 1988

スポーツゲーム
〈熱血高校ドッジボール部〉
■CZ-232AS
標準価格7,800円(税別)
©TECHNOS JAPAN CORP. 1988

アクションゲーム
〈パックマニア〉
■CZ-233AS
標準価格7,800円(税別)
©NAMCO

アクションゲーム
〈ニュージーランドストーリー〉
■CZ-230AS
標準価格8,800円(税別)
©TAITO CORP. 1989

スポーツゲーム
〈V'BALL〉
■CZ-246AS
標準価格7,900円(税別)
©TECHNOS JAPAN CORP. 1989

バイクレーシングゲーム
〈スーパーハングオン〉
■CZ-238AS
標準価格8,800円(税別)
©SEGA 1987

ジェットヘリ・シミュレーションゲーム
〈サンダーブレード〉
■CZ-239AS
標準価格9,500円(税別)
©SEGA 1987

アクションゲーム
〈ダウンタウン熱血物語〉
■CZ-254AS
標準価格8,800円(税別)
© TECHNOS JAPAN CORP. 1989

アクションゲーム
〈サイバリオン〉
■CZ-229AS
9月発売予定
©TAITO CORP. 1988

開発ツール

OS-9/X68000
■CZ-219SS 標準価格29,800円(税別)

Human68k ver2.0
■CZ-244SS 標準価格9,800円(税別)

THE福袋V2.0
■CZ-224LS 標準価格9,980円(税別)

AI-68K(Staff LISP/OPS PRO-68K)
■CZ-234LS 標準価格188,000円(税別)

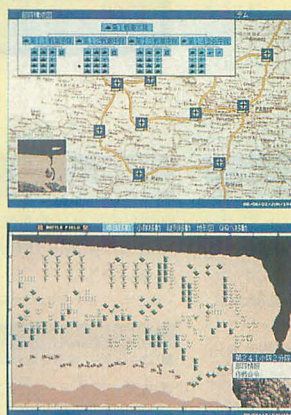
ARTDINK

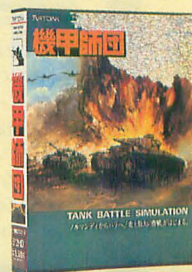
TANK BATTLE SIMULATION

機甲師団

X68000
9月28日発売

ノルマンディからパリへ。「史上最大の作戦」がはじまる。

難敵「ドイツ機甲師団」を撃破すべく
ついに総攻撃を開始した「連合軍第7機甲師団7大隊」。
戦車72両、2,000を超す将兵を擁し、壮絶なバトルを繰りひろげる。
師団司令部(GHQ)で映される、12もの戦場。
戦車ごと吹き飛ばされる「地雷原」、急斜面が待ちうける「山岳地帯」、
「市街地」「渡河」「要塞」など、手ごわい難局が待ちうける
戦況を読むうちに、ひらめく奇策。爆撃要請、兵員や物資の補給、
降伏、士官任命、一瞬のインスピレーションが戦局を大きく変える。
成功か、失敗か、貴官の判断が運命を決する。

●画面写真は開発中のものです。

X68000
●5"2HD(3枚組)
標準価格　**9,500円**

X68000 栄冠は君に　12月発売予定。

株式会社 アートディンク　〒275 千葉県習志野市津田沼2-11-20
TEL 0474-77-7541(ユーザサポート専用)

●お買い求めは、全国パソコンショップにて。
●通信販売(送料無料)をご希望の方は、住所・氏名・電話番号・商品名・機種名・メディア名を明記して左記まで9,500円と消費税3%を同封の上現金書留にてお申し込み下さい。

(表示価格に消費税は含みません)

地域	店名	電話
札幌	そごう電器YES 5F	(011)214-2850
	パソコンショップハドソン	(011)205-1590
青森	電巧堂チェーン青森本店	(0177)23-2356
盛岡	デンコードー盛岡本店	(0196)54-2772
秋田	電巧堂チェーン秋田駅前本店	(0188)34-3151
仙台	デンコードー仙台本店	(022)225-5290
	デンコードーDac仙台東口店	(022)291-4744
	庄子デンキコンピュータ中央店	(022)224-5591
郡山	うすい百貨店	(0249)32-0001
福島	庄子デンキ粉又駅前店	(0245)21-2011
いわき	いわきマイコンショップ	(0246)23-0513
山形	庄子デンキ山形七日町本店	(0236)42-1222
新潟	PIC	(025)243-5135
	PICこばり店	(025)233-5791
長岡	丸専デパート 1F	(0258)33-4970
	丸専マイコンショップジャスコ長岡店	(0258)27-6033
宇都宮	K&P宇都宮	(0286)32-0002
前橋	パソコンランド21前橋店	(0272)21-2721
高崎	パソコンランド21高崎店	(0273)26-5221
太田	パソコンランド21太田店	(0276)45-0721
伊勢崎	パソコンランド21伊勢崎店	(0270)21-3121
水戸	川又書店駅前店 2F	(0292)31-0102
つくば	ミドリ電気館	(0298)55-3715
大宮	ダイエー大宮店 6F	(0486)45-4147
川越	長崎屋川越新宿電気館 1F	(0492)44-5461
上尾	ボンベルタ上尾	(0487)73-8711
松戸	イトーヨーカドー松戸店	(0473)68-5131
千葉	ラオックス千葉店(PERIE 3F)	(0472)27-5318
八千代台	ラオックス八千代台店	(0474)86-2261
秋葉原	丸善無線ECCS 4F	(03) 255-4911
	ミナミ電気館 4F	(03) 255-4040
	サトームセン(メディアセンター)	(03)5256-3267
	サトームセン(ラジオ館) 5F	(03) 251-1464
	サトームセン本店 5F	(03) 253-5879
	CVA秋葉原 2F	(03) 258-3711
	コム本店	(03) 251-1523
	ザ・コンピュータ館	(03)5256-3111
新宿	ラオックス新宿店	(03) 988-8686
池袋	ビックカメラ池袋東口本店 4F	(03) 988-0020
	ビックカメラ池袋北口店 4F	(03) 988-8666
渋谷	上新電機J&P渋谷店 1F	(03) 496-4141
	ビックカメラ渋谷店A館 6F	(03) 477-0002
世田谷	ヤマギワ玉川店(高島屋 5F)	(03) 708-1638
武蔵野	ラオックス吉祥寺店 2F	(0422)21-3471
小金井	サン家電小金井店	(0423)85-3810
立川	丸善無線立川店(WIll 6F)	(0425)27-6211
八王子	ムラウチ電気 2F	(0426)42-6211
	J&P八王子店(そごう 7F)	(0426)26-4141
昭島	イトーヨーカドー 3F	(0425)46-1411
町田	東急ハンズ町田店 B1F	(0427)28-2603
	J&P町田店	(0427)23-1313
横浜	ソフトクリエイト横浜店	(045)314-4777
	横浜VIVRE21 7F	(045)314-2121
	ダイエー戸塚店 3F	(045)881-1261
	ICコスモランドあざみの	(045)901-1901
	ICコスモランドかもい	(045)934-9636
川崎	セキグチデンキ館	(044)244-5421
藤沢	WAVE EYE	(0466)43-1771
平塚	梅鳴屋	(0463)22-4147
厚木	ラオックス厚木店オーディオ館	(0462)22-2722
大和	WAVE EYE大和店	(0462)63-9898
甲府	中込電気商会	(0552)24-5431
長野	ダイエー長野店 7F	(0262)27-1311
松本	還兵	(0263)32-6350
金沢	うつのみや片町店	(0762)21-6136
富山	丸の内カラー駅前店	(0764)41-9075
福井	PASわくわくメディア館	(0776)23-7621
	だるまや西武 7F AVC	(0776)27-0111
沼津	メルパ沼津	(0559)22-4858
静岡	メルパ静岡	(0542)54-5338
富士	メルパ富士店	(0545)51-8022
浜松	メルパ浜松本店	(0534)54-8412
	ホーエー家電有楽店	(0534)53-1441
	マエダM&V	(0534)73-1691
名古屋	栄電社テクノ名古屋	(052)581-1241
	パソコンショップコムロード	(052)263-5828
	ちくさ正文館書店ターミナル店	(052)732-3601
	カトー無線電機電気館 4F	(052)264-1534
	丸善無線 第一アメ横店	(052)263-1626
	栄電社テクノ大須	(052)252-0181
	トップカメラ 4F	(052)971-0111
	J&P大須店	(052)262-1141
	エイデンメディア大須	(052)242-8591
豊橋	栄電社テクノ豊橋	(0532)52-1231
岡崎	ジャスコ岡崎店 3F	(0564)23-4960
豊田	ジャスコ豊田店パソコンショップJPC	(0565)35-1761
岐阜	パソコンショップコムロード岐阜店	(0582)66-0286
大垣	スイテック大垣ヤナゲン店A館 6F	(0584)81-3491
津	Kawai無線OA 津店	(0592)26-0111
四日市	Kawai無線カデンワールド四日市	(0593)54-3366
松阪	Kawai Bax店 6F	(0598)26-0111
大津	西武百貨店関西大津店	(0775)25-0111
日本橋	J&Pテクノランド	(06) 634-1211
	J&Pメディアランド	(06) 634-1511
	ニノミヤエレランド	(06) 632-2038
	ニノックスコア日本橋店	(06) 647-2038
難波	ニノミヤパソコンランド難波店	(06) 643-3217
梅田	ニノミヤパソコンランド駅前第4ビル店	(06) 341-2031
	J&P阪急三番街店	(06) 374-3311
高槻	上新電機J&P高槻店	(0726)85-1212
八尾	西武百貨店関西八尾店	(0729)97-0111
京都	J&P京都寺町店	(075)341-3571
和歌山	J&P和歌山店	(0734)28-1441
神戸	星電社三宮本店C-ソフト	(078)391-6171
	ダイエー三宮電器館パレックス	(078)391-7911
	上新電機三宮一番館	(078)211-2111
姫路	上新電機J&P姫路店	(0792)22-1221
	姫路コンピュータソフトコンパックス	(0792)94-8244
岡山	岡山VIVRE21	(0862)32-6881
倉敷	倉敷ファソランド	(0864)25-4701
広島	松本無線パーツ	(082)243-4451
	ダイイチパソコンCITY	(082)248-4343
高松	Sound Check MOVE	(0878)61-6171
徳島	徳島そごう 7F	(0886)25-4056
高知	デンキのタグチ	(0888)23-0181
松山	ダイイチ松山パソコンCITY	(0899)31-6711
北九州	ベストマイコン小倉パソコン館	(093)551-6281
福岡	ベストマイコン福岡店	(092)781-7131
	寿屋エレデ博多	(092)281-4411
長崎	ベスト電器長崎新地店	(0958)28-1333
熊本	寿屋本荘店	(096)372-5411
	ベストマイコン熊本パソコン館	(096)332-4180
鹿児島	ベスト電器鹿児島パソコン館	(0992)23-2081
大分	ベスト電器大分パソコン館	(0975)32-9336
宮崎	宮崎寿屋百貨店 7F	(0985)27-4111
那覇	ベスト電器那覇店 6F	(0988)62-7588

通信販売をご希望の方、ソフト名・機種名・住所・氏名・電話番号を明記の上、TAKERU事務局まで必ず現金書留でお申し込み下さい。(送料は無料です)

ブラザー工業株式会社

〒467 名古屋市瑞穂区堀田通 9-38
新事業推進室
TAKERU事務局(052)824-2493
東京営業所(03)274-6916
大阪営業所(06)252-4234

※コンピューター・ソフトウェアは著作物です。著作者の許可なくコピーあるいはレンタル等の行為は、法律で禁じられています。

HYDLIDE 3 SPECIAL VERSION

© 1989,1990 T&E SOFT

10月13日発売!

究極のハイドライド、タケルよりX68000に登場!!

コンピュータゲーム業界に新境地を開いたA.R.P.G.ハイドライドシリーズ。話題をさらったハイドライド1、2。絶賛のゲームシステム、ハイドライド3。さらにエキサイティングにバージョンアップしたハイドライド3SV(スペシャルバージョン)。こうした華麗なる進化を経て、この秋、究極のスペシャルバージョン「ハイドライド3 SV X68000版」が登場!X68000の特性をフルに活かし、よりクリアに、よりピュアになったフェアリーランドが冒険心を呼び覚ます。

★城、街、ダンジョン、200階建ての
空中都市から未知の空間へ、さらに
の領域が増えて、驚異のパラレルワ
ルドはより広域化。

★ソフトウェア3プライオリティーの
全重ね合わせと、群を抜いた美しい
ラフィックス。

★時間の概念を導入し、時刻によって
景の色合いが変わる。

★新たなキャラクターと新たなマップ
アニメーション導入が"SV"たら
めた。

X68000
¥7,200

©T&E SOFT

闇の血族

魅由の繰り広げる
ミステリアスアニメーションアドベンチャー第2弾!!

下巻

NOVEL WARE

THE PREDESTINED HOMICIDES #2
完結編

密林深く眠る失せし文明に、蘇る血の運命(さだめ)。すべての謎は、一人の少女の下に今、遥かなる真実を紡ぎ出す。

1990年6月
古えの封印は解かれ、時間(とき)の糸車は無数の運命の糸を引き、血染めの歴史絵を紡ぎはじめた。一連の殺人事件に秘められた暗号を解く唯一の手掛かりを求めて、魅由は親友の理沙と共に中米の地へとおもむく。が、そこには思いもよらぬ宿命の罠が待ち受けていたのであった。

下巻9月新発売

華麗さと迫力でせまるアニメシーン

神秘的なまでに美しい遺跡

このまま、生け贄とされてしまうのか……

X68000対応 5"-2HD
●ローランド社
MT32、CM32L、CM64完全対応
MIDIインターフェイスボードC-Z-6BMI
又は、SACOM製SX-68Mが必要です。
標準価格 8,800円

※標準価格には消費税は含まれておりません。

上巻好評発売中!!

SACOM
株式会社 システム サコム
〒130 東京都墨田区両国4-38-16
両国桜井ビル4F
ハードウェア部 03(635)5145
ソフトウェア部 03(635)7609

X68000でのソフトが遊べるぞ!! 第一弾はだっ!!

élf エ・ル・フ

PINKY・PONKY①〜③(ぴんきい・ぽんきい)

主人公であるあなたは、究極のナンパ師。彼女とうまく言葉をかわし、彼女の機嫌を損なわないよう頭を使ってくどき落してください。しかし、ただ単純にコマンドを入力しても決してハッピーエンドにはなりません。彼女をものにしてお楽しみができるかどうかは、まさにキミの腕次第！コマンド選択方式、ナンパゲームの最高峰。美少女ソフトの常識を卓越した、ロイヤルビューティフル・グラフィックは、全体の3分の2を占める大画面。豊富で痛快なリアクション(1シーン2万文字!!)が2人の会話を盛り上げます。もちろん、アニメーション(全機種)FM音源によるBGM(MSX版は除く)が随所に挿入されています。あなたの名前を登録する事ができますので、よりリアルにストーリーが展開されます。
ぴんきい・ぽんきい はナンパソフトの決定版です!!

DE・JA (デジャ)

画面全体の2／3を占めるビックなサイズのグラフィックが１７０画面。膨大なメッセージ、あなたの頭脳に挑戦する数々のトリック。もちろん美少女達もからみにからみ、燃えに燃える。美少女ソフトの横綱、エルフがお届けする本格的アドベンチャーゲーム。これをやらずに90年代は語れない!？

9月下旬！　定価6,800円

ドラゴン・ナイト

大変お待たせ致しました!!美少女ソフトの金字塔、「ドラゴンナイト」の登場です。女の王国「ストロベリーフィールズ」。その王国は神聖なる女神の恩恵を受け、人々は平和に暮らしていた。しかしある日、その女神が住む塔の上空に不気味なドラゴンの影が…。そしてその国に立ち寄った勇敢なる、剣士「タケル」運命は…。

10月下旬！　定価6,800円

⇨好評発売中!!

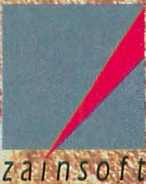
各定価5,800

エルフでは、すでに発売されているソフト。これから発
される全てのソフトをX68000に移植する事が決定致
ました。エルフではユーザーの皆さまに、いかに楽しん
いただくかをモットーに一生懸命がんばります。どうぞ
期待ください!!

通信販売をご希望の方は…

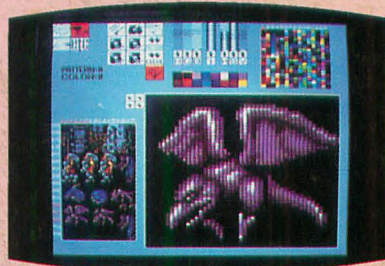
●現金書留の場合……
商品名、機種、メディアを明記の上エルフまでお送り下さい。
●郵便振替の場合……
郵便局の振替用紙に商品名、機械、メディアを明記の上、口座番号
東京3-191196
エルフまでお申し込み下さい。

株式会社 エ・ル・フ
〒169 東京都新宿区北新宿1-12

zainsoft

ザインの彩先端。

ニューコンセプトのアートキャンバス「G=ツール」、登場。

オリジナルアートが驚くほど自由に描ける、X68アーティストの感性を刺激するアドバンスト・グラフィックツール『G＝ツール』。作品に挑む上で、必要不可欠なグラフィックキャラクタ、背景作成のすべてを備えたトータルツールです。これまで3回にわたってその特長を説明してまいりましたが、今回リリースに際し、もう一度驚くべきパフォーマンスの概要をご紹介します。

新発売
¥28,000

G・ツール

GR EDITモード

マルチウインドウシステム
最高12枚まで描画ウインドウが開けます。

ユーザーアイコンシステム
使い勝手に合わせて、自分流のアイコンボードが作成可能。

マウス機能定義システム
マウスの左右に機能定義が可能。

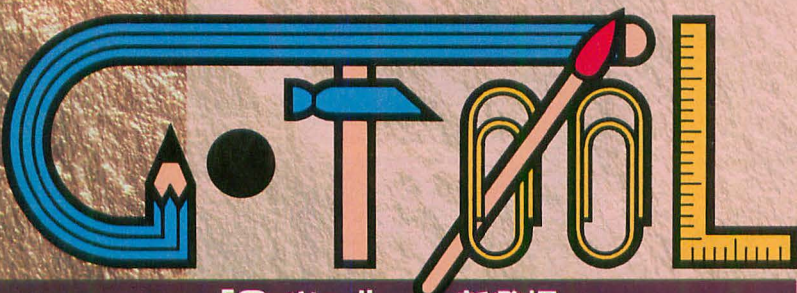
高速メニューウインドウ処理
メニューウインドウの開閉が瞬時に行えます。

BG EDITモード

スプライト処理
作成から修正、アトリビュートが行えます。

スプライトカラー処理
16ページ分まとめて処理できます。

背景の作成
最大250画面分を自由に設定することができます。

キャラクタチェック機能
単独チェックのほか、背景と重ねてのチェックも可能。

zainsoft

株式会社ザインソフト 〒651 兵庫県神戸市中央区磯辺通2丁目2-10 新南泰ビル10F TEL.(078)242-2855

資料・オートデモ
請求券
OH!X 10月号

neXt
RPG・ACT・SLG、最強のラインナップで
次世代体験……… neXt！

UNDEADLINE

「幻獣鬼」が今発揮する。

ステージセレクションシステム＋転職システムをとり入れたアクションゲームの決定版！

X68000 ACT-neXt 幻獣鬼

鋭い！ X68000ユーザーの鋭き感性をより研ぎ澄ます
鋭い！ 魂より出ずる鋭き野望が渦巻く世界
鋭い！ プレイヤーの鋭きテクニックがすべての明暗を分かつ

●X68000 5"2HD 3枚組
標準価格 ¥7,800
※表示価格に消費税は含みません。

X68000の本質。「黒衣の貴公子」が今解き明かす。

黒衣の貴公子

経験値を捨てた！ 新世代マルチストーリーロールプレイングゲーム

X68000 RPG-neXt 黒衣の貴公子（ルーンワース）

熱い！ X68000ユーザーの熱き要望に応え堂々登場!!
熱い！ ルーンワースとよばれる異世界で繰り広げられる熱き冒険譚
熱い！ プレイヤーの熱き魂が物語を自由に織りなしてゆく

●X68000 5"2HD 3枚組
- 全グラフィック描き起こし（高解像グラフィック 512×512ドット）
- ジョイスティック対応
- FM音源8音＋ADPCM音源対応

●PC-9801VM、UVシリーズ PC-286、386シリーズ、NOTE対応 5"2HD/3.5"2HD 2枚組 ●サウンドボード対応 ●ジョイスティック対応
※要640Kバイト
※PC-9801/E/F/M/VF/U2/XA、PC-286L/LE/LFおよびPC-286NOTE Executiveでは、ドライブ、RAM等の増設の如何にかかわらず、作動いたしません。

●PC-8801SRシリーズ・VA、98DO対応 5"2D 5枚組

● MSX2/MSX2+ (RAM64K以上、VRAM128K以上) 3.5"2DD 3枚組

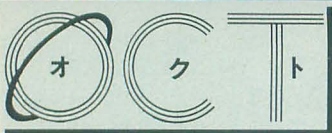
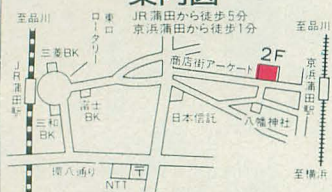
標準価格 各¥8,800
※表示価格に消費税は含みません。

■通信販売ご希望の方は現金書留で料金と商品名・機種名と電話番号を明記の上、当社迄お送りください（速達希望の方は300円プラス）
■カタログご希望の方は、送料として切手200円分を同封の上、カタログ請求券をお送りください（葉書での請求はお断わりします）
●T&Eの最新情報がわかるテレフォンサービス 名古屋(052)776-8500

T&E SOFT
企画・開発・製造・販売
株式会社 ティーアンドイーソフト
〒465 名古屋市名東区豊が丘1810番地 PHONE:052-773-7770

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 オクトハッピークレジットをご利用下さい!!

■朗報です。冬のボーナス一括払い(12月末)OK!!手数料なし。(翌月末払いもOK!!)ご利用下さい。

パソコンプラザ

'90オクトで始まるパソコンワールド

☎03-730-6271

●営業時間 AM 11:00～9:00/日曜・祭日PM7:00　電話一本で、ハイ即納

〒144 東京都大田区蒲田4-6-7 FAX 03-730-6273

全国通販

●定休日毎週火曜日 祭日の場合翌日になります。

オクト ラクラククレジット

1回	2%	3回	2.5%	6回	3.5%	10回	5%	12回	5%	15回	7.5%
18回	9%	20回	10%	24回	11%	30回	14.5%	36回	15.5%	48回	20%

OCT-1 システム インフォメーション

- ▶全商品保証付(メーカー保証)
- ▶超低金利ハッピークレジット(1回～60回)頭金ナシOK!
- ▶ボーナス一括払いOK!ボーナス2回払いOK!!
- ▶配達日の指定OK!(万全なサポート体制)
- ▶商品の組合せ自由! オクトフリーダムシステム
- ▶店頭デモンストレーション実施中

オクト セレクテッドシステム

広告掲載商品以外の製品も取扱っております。

蒲田

●冬のボーナス一括払い(12月末)OK!!
手数料なしです。絶対、お得ですゾ。
翌月末払いも受付けています(10月末)

★下記セットでお買い上げの方にはプレゼント!! ●①MD-2HD 10枚 ②ジョイカード 2個(連射式) ③シリコンキーボードカバー ④ゲームソフト サンダーブレード(￥9500)

お好みのセットをお選び下さい。　送料無料

●SX-WINDOW搭載。
●40Mバイトハードディスク搭載

EXPERTⅡ・EXPERTⅡ-HD

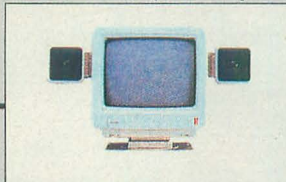
●CZ-603C-BK/GY
定価￥338,000
●CZ-613C-BK/GY
定価￥448,000

現金特価!! 推選
お電話下さい。

●SX-WINDOW搭載。
●拡張I/Oポート4スロット装備

PROⅡ・PROⅡ-HD

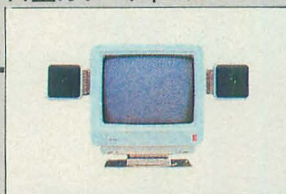
●CZ-653C-BK/GY
定価￥285,000
●CZ-663C-BK/GY
定価￥395,000

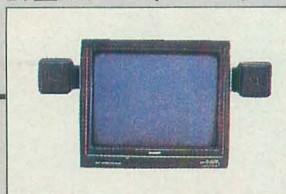
CZ-8NJ2 限定
●インテリジェントコントローラ
定価￥23,800
超特価￥18,600

15型カラーディスプレイTV

CZ-605D-GY/BK
定価￥115,000

15型カラーディスプレイTV

CZ-613D-GY/BK
定価￥135,000

14型カラーディスプレー

CZ-604D-GY/BK
定価￥94,8000

21型カラーディスプレイ

CU-21HD
定価￥148,000

Ⓐ CZ-603C＋CZ-605D…………定価合計￥453,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓑ CZ-613C＋CZ-605D…………定価合計￥563,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓒ CZ-653C＋CZ-605D…………定価合計￥400,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓓ CZ-663C＋CZ-605D…………定価合計￥510,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓔ CZ-603C＋CZ-613D…………定価合計￥473,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓕ CZ-613C＋CZ-613D…………定価合計￥583,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓖ CZ-653C＋CZ-613D…………定価合計￥420,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓗ CZ-663C＋CZ-613D…………定価合計￥530,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓘ CZ-603C＋CZ-604D…………定価合計￥429,800▶オクト大特価

12回	￥28,000	24回	￥14,800	36回	￥10,200	48回	￥8,000

Ⓙ CZ-613C＋CZ-604D…………定価合計￥542,000▶オクト大特価

12回	￥36,000	24回	￥19,000	36回	￥13,100	48回	￥10,200

Ⓚ CZ-653C＋CZ-604D…………定価合計￥379,800▶オクト大特価

12回	￥25,400	24回	￥13,400	36回	￥9,300	48回	￥7,200

Ⓛ CZ-663C＋CZ-604D…………定価合計￥489,800▶オクト大特価

12回	￥32,200	24回	￥17,000	36回	￥11,800	48回	￥9,200

Ⓜ CZ-603C＋CU-21HD…………定価合計￥486,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓝ CZ-613C＋CU-21HD…………定価合計￥596,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓞ CZ-653C＋CU-21HD…………定価合計￥433,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

Ⓟ CZ-663C＋CU-21HD…………定価合計￥543,000▶オクト大特価

12回	?	24回	?	36回	?	48回	?

♡どんどんTELしよう。安くなるかもヨ!!

♡クレジット価格は、消費税込みですヨ。ご利用下さい!!

■便利です 夜9時まで営業しております お立寄り下さい お待ちしております!!

※クレジットの回数は1回～60回、ボーナス併用などありますのでお電話でお問合せ下さい。

■本体セット:送料無料　●店頭デモ実施中…専門の係員が詳細にアドバイス致します。ぜひご来店下さい。

※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは、電話でお問合せ下さい。

■店頭にて、ゲームソフト25%OFF!!(税別)、超低金利 ハッピークレジットをご利用ください!!
■特に人気のある商品によっては、しばらくお待ち願うことがありますのでご了承下さい。

厳選された製品を、より安く、より早く、皆様のお手元に!!

広告掲載商品以外の
製品も取扱っております。

チャンス! X68000・SUPER-HD(チタン)=好評・発売中 どんどんTEL下さいネ。 送料ナシ!!

SX-WINDOW搭載。

●ザ・ワークステーションと呼ぶにふさわしい
スーパーな68000!! 新登場!!
SUPER-HD。

※プレゼント! ①MD-2HD10枚 ②サンダーブレード(¥9,500) ③ジョイカード(連射式) ④シリコンキーボード(¥2,800)

X68000 SUPER-HD

●CZ-623C-TN+CZ-613D-TN
定価合計¥633,000…大特価!! TEL下さい。

※マウス・トラックボール付!! ディスプレイにはスピーカ2個、チルト台付!!

12回	?	24国	?	36回	?	48回	?

他のディスプレイ① CZ-602D、② 612D、③ CZ-603D、④ CU-21HDの組合せもございますのでお問い合せ下さい。

♡安くてゴメンなさい。今だけヨ!!

※超低金利クレジットご利用下さい。1回〜60回払い、頭金ナシ! ボーナス1回払い、ボーナス2回払いOK!

X68000 EXPERT-HD

オクト限定スペシャルセット

ラストチャンス!! 早い者勝ち!!

●CZ-612C(BK)(¥466,000)
●CZ-602D(BK)(¥99,800)
●MD-2HD 10枚
●ジョイカード(連射式×2個)
●シリコンキーボード・カバー

オクト超特価
¥364,000(送料・消費税込み!!)

※ディスプレイ=①CZ-604D ②CZ-605D ③CZ-613D ④CU-21HD
との組合せもございます。TEL下さい。

オクト特選 シャープ周辺機器 (送料¥1,000)

●CZ-6BE1 1MB増設RAMボード……(¥ 35,000)▶特価¥ 26,500
●CZ-6BE1B 1MB増設RAMボード………(¥28,000)▶特価¥21,000
●CZ-6BE2 2MB増設RAMボード……(¥ 79,800)▶特価¥ 60,500
●CZ-6BE4 4MB増設RAMボード……(¥138,000)▶特価¥104,800
●CZ-6BF1 増設用RS-232Cボード……(¥ 49,800)▶特価¥ 38,500
●CZ-6BG1 GP-IBボード……(¥ 59,800)▶特価¥ 45,000
●CZ-6BM1 MIDIボード……(¥ 26,800)▶特価¥ 20,500
●CZ-6BN1 スキャナ用パラレルボード‥(¥ 29,800)▶特価¥ 22,800
●CZ-6BP1 数値演算プロセッサボード(¥ 79,800)▶特価¥ 60,500
●CZ-6BO1 ユニバーサルI/Oボード…(¥ 39,800)▶特価¥ 30,500
●CZ-6EB1/BK 拡張I/Oボックス……(¥ 88,000)▶特価¥ 66,800
●CZ-6VT1/BK カラーイメージ・ユニット…(¥ 69,800)▶特価¥ 53,000
●CZ-6BL2 LANボード……(¥298,000)▶大 特 価
●CZ-8NM2A マウス……(¥ 68,800)▶特価¥ 5,300
●CZ-8NT1 マウストラックボール‥(¥ 98,800)▶特価¥ 7,500
●CZ-8NS1 カラーイメージスキャナ……(¥188,000)▶ 大 特 価
●CZ-6BC1 FAXボード……(¥ 79,800)▶特価¥60,500
●CZ-8TM2 モデムユニット……(¥ 49,800)▶特価¥38,000
●CZ-64H 増設ハードディスク…(¥120,000)▶ 大 特 価
●CZ-6TU GY/BK RGBシステムチューナー…(¥ 33,100)▶特価¥25,000
●BF-68PRO 高性能CRTフィルター……(¥ 19,800)▶特価¥15,500
●SX-68M(システムサコム) MIDIボード……(¥ 19,800)▶特価¥15,000
●PIO-68BE1-A(I/O DATA) 1MB増設RAMボード……(¥ 25,000)▶特価¥18,500
●PIO-6BE2-2M(I/O DATA) 2MB増設RAMボード……(¥ 50,000)▶特価¥37,000
●PIO-6BE4-4M(I/O DATA) 3MB増設RAMボード……(¥ 88,000)▶特価¥65,000
●CZ-6BV1 ビデオボード……(¥ 21,000)▶特価¥15,800

オクト面白グッズ

アイテック(送料¥1,000)
●IT-X640(¥158,000)…………特価¥103,000
●IT-X680(¥198,000)…………特価¥134,000

モデムコーナー(送料¥1,000)

●MD-1200AIII……特価¥14,800
●MD-24FS4……特価¥31,500
●MD-24FS5……特価¥34,800
●MD-24FP4……特価¥27,900
●MD-12FS……特価¥15,000

熱転写カラー漢字プリンター (ケーブル 用紙付) 送料¥1,000

CZ-8PC4 ¥99,800 限定
●48ドット サーマルヘッド
●B5〜B4まで
●ハガキ可能
●カラー対応
オクト大特価 ¥64,800

①CZ-8PK10(24ピン漢字プリンター136桁)
定価¥97,800……大特価!! TEL下さい。
②CZ-8PG1(24ピンカラー漢字プリンター80桁)
定価¥130,000……大特価!! TEL下さい。
③CZ-8PG2(24ピンカラー漢字プリンター136桁)
定価¥160,000……大特価!! TEL下さい。
④IO-735×(カラーイメージジェット)
定価¥248,000……大特価!! TEL下さい。

パソコンラック 推奨 送料無料

①五段キャスター付

5段キャスター付
キーボードが収納できるから、手元でマウス操作がラクラクできる
棚板5段のマルチに活用できるディスク。
ウーン、こいつはデキル!
1325(H)×640(W)×700(D)
特価¥16,000

②四段キャスター付

4段キャスター付
どんなパソコンにもフレキシブルに対応! 使い易いデスクです。
1245(H)×614(W)×600(D)
特価¥12,000

③三段キャスター付

3段キャスター付
場所を選ばない簡易で便利なディスクです。
限定
1175(H)×640(W)×600(D)
特価¥8,800

X68000ソフト大セール実施中※ゲームソフトオール25%off

〈グラフィック〉●Z's STAFF PRO68K Ver.2.0 (シャフト)定価¥58,000 オクト特価¥40,000
〈データーベース〉●KAMIKAZE (サムシンググッド)定価¥68,000 オクト特価¥46,000
〈グラフィック〉●C-TRACE68 (キャスト)定価¥68,000 オクト特価¥51,000
〈C言語〉●C & Professional Pack (マイクロウェアジャパン)定価¥58,000 オクト特価¥44,000
〈グラフィック〉●サイクロン エキスプレス 定価¥78,000 オクト特価¥58,000
〈グラフィック〉●デジタルクラフト 定価¥39,800 オクト特価¥28,000
〈ワープロ〉●ハイパーワード 定価¥39,800 CZ-251BS オクト特価¥29,800

型 名	商 品	定 価	特 価
CZ-211LS	Ccompiler PRO-68K	¥39,800	¥28,800
CZ-212BS	BUSINESS PRO-68K	¥68,000	¥48,000
CZ-213MS	MUSIC PRO68K	¥18,800	¥13,500
CZ-214MS	SOUND PRO-68K	¥15,800	¥11,500
CZ-215MS	Sampling PRO-68K	¥17,800	¥12,800
CZ-219SS	OS-9/X68000	¥29,800	¥21,000
CZ-220BS	DATA PRO-68K	¥58,000	¥41,000
CZ-257CS	Print Shop PRO68K.V.2	¥19,800	¥14,300
CZ-223CS	Communication PRO-68K	¥19,800	¥14,300
CZ-224LS	THE 福袋 V2.0	¥ 9,900	¥ 7,500
CZ-226BS	CARD PRO-68K	¥29,800	¥21,300
CZ-241BS	システム手帳リフィル集	¥ 9,800	¥ 7,500
CZ-242BS	活用フォーム集	¥ 9,800	¥ 7,500
CZ-244SS	Homan 68K Ver.2.0	¥ 9,800	¥ 7,500
CZ-247MS	MUSIC PRO-68K〔MIDI〕	¥28,800	¥20,800
CZ-240BS	Stationery PRO-68K	¥14,800	¥11,500
CZ-243BS	CYBER NOTE PRO-68K	¥19,800	¥15,200
EW		¥38,000	¥29,800
G-68K		¥14,800	¥11,400
E-68		¥19,800	¥15,300

★オクト今月だけの新品限定販売(各1台限)(送料¥1,000)

●CZ-611C(BK)定価¥399,800……大特価¥218,000
●CZ-652C(BK)定価¥298,000……大特価¥188,000
●CZ-662C(BK)定価¥408,000……大特価¥248,000
●CZ-601D(BK)定価¥119,800……大特価¥ 68,000
●CZ-601D(GY)定価¥119,800……大特価¥ 68,000
●CZ-612D(GY)定価¥119,800……大特価¥ 74,000
●IO-735 定価¥248,000……大特価¥158,000

店頭ゲームソフトオール25%off! ビジネスソフト 25%より特価中

●尚、送料として1ケ¥500、2ケ¥700、3ケ以上で¥1,000となります。(税別)

★通信販売お申込みのご案内★ 〒144 東京都大田区蒲田4-6-7 TEL:03-730-6271

お申込みはお電話でお願いします。お客様の〈住所〉〈氏名〉〈電話番号〉及び〈商品名〉をお知らせ下さい。●入金確認後ただちに商品をご送付いたします。

現金一括払い
銀行振込:お近くの銀行より(電信扱い)にてお振込み下さい。
現金書留:封筒の中に住所・氏名・商品名をご記入の上当社までお送り下さい。

クレジット
専用お申込用紙をお送り致します。ので、必要事項をご記入、ご捺印の上ご返送下さい。手続きは簡単です。

オクト ラクラク クレジット表

1回	2%	3回	2.5%	6回	3.5%	10回	5%
12回	5%	15回	7.5%	18回	9%	20回	10%
24回	11%	30回	14.5%	36回	15.5%	48回	20%

振込先
富士銀行 久ヶ原支店 ㊟No.1824
三菱銀行 蒲田支店 ㊟No.0278691
株式会社 億人(オクト)

※掲載の価格は変動しますので、まずは、お電話にてご確認ください。
※連休のお知らせ=
※上記料金には、消費税は含まれておりません。消費税が付加されますので、詳しくは電話でお問合せ下さい。
※銀行振込、または、現金書留でご注文の際には、あらかじめ電話でご確認の上、お申し込み下さい。

■平成2年冬のボーナス一括払いOK!!(12月末)手数料ナシ!!
超低金利クレジットをご利用下さい。

超低金利クレジットO

注目!!

《翌月一括払い(10月末)(11月末)はもちろん》
冬のボーナス一括払い
手数料(金利)無料
(平成2年12月末払いをご利用下さい。)

またまた秋葉原で おなじみの

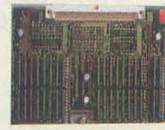
9/15〜10/15

●お近くの方はお
●本体単品で特
●ビジネスソフト定

プリンター 10台限定 (送料¥1,000)
■CZ-8PK8 (定価¥152,000)
●24ピン漢字プリンタ(136桁)
●ハガキ印字OK!!
P&A限定特価¥49,800 (送料・消費税込¥52,324)

CYBER STICK
●CZ-8NJ2
(定価¥23,800)
超特価!!
¥18,500 (送料・消費税込み¥19,570)

X68000シリーズ専用 特価¥16,480
MIDIインターフェースボード
SX-68M (サコム)
(純生コンパチ)定価¥19,800
送料・消費税込み!

X68000用メモリーボード (I/O・DATA) (送料¥500)

①PIO-6BE1-A 定価¥25,000 …… ¥18,000 (送料・消費税込¥19,055)
②PIO-6BE2-2M 定価¥50,000 …… ¥36,500 (送料・消費税込¥38,110)
③PIO-6BE4-4M 定価¥88,000 …… ¥64,300 (送料・消費税込¥66,744)

ジョイスティック 送料¥500
●X-1PRO 定価¥9,500▶特価¥7,800
●ASCII STICK 定価¥6,800▶特価¥5,500

P&A超低金利クレジットをご利用ください!!

NEW X68000 EXPERT II/II-HD & PRO II/PRO II-HD & SUPER-HD (送料・消費税込)

EXPERT II

セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

EXPERT II

セット	定価	特価	12回	24回	36回	48回	60回
Ⓐセット:CZ-603C+CZ-604D	¥432,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット:CZ-603C+CZ-605D	¥453,000	(価格はお電話下さい。)	30,200	15,900	11,000	8,500	7,100
Ⓒセット:CZ-603C+CZ-613D	¥473,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット:CZ-603C+CU-21HD	¥486,000	(価格はお電話下さい。)	?	?	?	?	?

EXPERT II-HD

セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

EXPERT II-HD

セット	定価	特価	12回	24回	36回	48回	60回
Ⓐセット:CZ-613C+CZ-604D	¥542,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット:CZ-613C+CZ-605D	¥563,000	(価格はお電話下さい。)	37,700	19,800	13,700	10,600	8,900
Ⓒセット:CZ-613C+CZ-613D	¥583,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット:CZ-613C+CU-21HD	¥596,000	(価格はお電話下さい。)	?	?	?	?	?

PRO II

セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

PRO II

セット	定価	特価	12回	24回	36回	48回	60回
Ⓐセット:CZ-653C+CZ-604D	¥379,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット:CZ-653C+CZ-605D	¥400,000	(価格はお電話下さい。)	26,800	14,100	9,700	7,600	6,300
Ⓒセット:CZ-653C+CZ-613D	¥420,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット:CZ-653C+CU-21HD	¥433,000	(価格はお電話下さい。)	?	?	?	?	?

PRO II-HD

セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

PRO II-HD

セット	定価	特価	12回	24回	36回	48回	60回
Ⓐセット:CZ-663C+CZ-604D	¥489,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット:CZ-663C+CZ-605D	¥510,000	(価格はお電話下さい。)	34,100	17,900	12,400	9,600	8,100
Ⓒセット:CZ-663C+CZ-613D	¥530,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓓセット:CZ-663C+CU-21HD	¥543,000	(価格はお電話下さい。)	?	?	?	?	?

■P&A恒例サマー大バーゲン祭開催中!!

◎電話にて、ドンドンお問合せ下さい!!
クレジット表には、出せないほどの価格です。
メーカーさん、ご免なさい。
ユーザーの方には大歓迎されそうです。
今がチャンスです、ハイ。

SUPER-HD

セットでお買い上げの方に、
●ディスケット10枚
●ジョイカード2ケ
プレゼント中!!

SUPER-HD

セット	定価	特価	12回	24回	36回	48回	60回
Ⓐセット:CZ-623TN+CZ-604D	¥592,800	(価格はお電話下さい。)	?	?	?	?	?
Ⓑセット:CZ-623TN+CZ-605D	¥613,000	(価格はお電話下さい。)	?	?	?	?	?
Ⓒセット:CZ-623TN+CZ-613D	¥633,000	(価格はお電話下さい。)	42,700	22,500	15,500	12,100	10,100
Ⓓセット:CZ-623TN+CU-21HD	¥646,000	(価格はお電話下さい。)	?	?	?	?	?

X68000シリーズ 〜P&Aスペシャルセット=限定誌上販売!!

台数限定　送料、消費税込み
セットでお買上げの方に、
●ディスケット10枚 ●ジョイカード2個 プレゼント中

EXPERT

- CZ-602C+CZ-612D ……定価¥475,800▶特価¥306,000
- CZ-602C+CZ-604D ……定価¥450,800▶特価¥300,000
- CZ-602C+CZ-605D ……定価¥471,000▶特価¥320,000
- CZ-602C+CZ-613D ……定価¥491,000▶特価¥336,000
- CZ-602C+CU-21HD ……定価¥504,000▶特価¥338,000

EXPERT-HD

- CZ-612C+CZ-612D ……定価¥585,800▶特価¥375,000
- CZ-612C+CZ-604D ……定価¥560,800▶特価¥369,000
- CZ-612C+CZ-605D ……定価¥581,000▶特価¥386,000
- CZ-612C+CZ-613D ……定価¥601,000▶特価¥403,000
- CZ-612C+CU-21HD ……定価¥614,000▶特価¥407,000

PRO-HD

- CZ-662C+CZ-612D ……定価¥527,800▶特価¥339,000
- CZ-662C+CZ-604D ……定価¥502,800▶特価¥333,000
- CZ-662C+CZ-605D ……定価¥523,000▶特価¥352,000
- CZ-662C+CZ-613D ……定価¥543,000▶特価¥368,000
- CZ-662C+CU-21HD ……定価¥556,000▶特価¥372,000

●本広告の掲載の商品の価格については、消費税は含まれておりません。●営業時間=平日AM10:00〜PM7:00、日祭AM10:00〜PM6:00

～84回払いまでOK.!!

★頭金なし!★即日発送

●価格は流通事情により変動致しますので、銀行振込・書留等の送付前に、あらかじめお電話にてご確認下さい。

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

超特価でクレジットが組める!!

X68000用ソフトコーナー(送料1ヶ～5ヶまで¥500)

商品	定価	特価
Z's STAFF PRO68K Ver2.0(ツァイト)	¥58,000	¥39,500
Z's TRIPHONY デジタルクラフト(ツァイト)	¥39,800	¥27,500
テラッツォ(ハミングバード)	¥19,800	¥15,840
KAMIKAZE(サムシング・グッド)	¥68,800	¥45,500
EW&EI(イースト)	¥38,800	¥29,500
C&Professional Pack(マイクロウェアジャパン)	¥58,800	¥43,000
Final Ver3.2(エーエスピー)	¥38,000	¥30,400
DATA PRO68K CZ220BS	¥58,000	¥40,500
CARD PRO68K CZ226BS	¥29,800	TEL下さい。
C compiler PRO68K CZ211LS	¥39,800	¥28,500
OS-9/X68000 CZ219SS	¥29,800	¥20,700
AI-68K CZ234LS	¥188,000	TEL下さい。
THE 福袋 V2.0 CZ224LS	¥9,900	¥7,400
SOUND PRO68K	¥15,800	¥11,300
MUSIC PRO68K CZ213MS	¥15,800	¥13,300
Sampling PRO68K CZ215MS	¥17,800	¥12,500
MUSIC-studio PRO68K 237MS	¥15,800	TEL下さい。
MUSIC-PRO68K〔MIDI〕247MS	¥28,800	¥20,500
New-print Shop 221HS	¥19,800	TEL下さい。
Communication 223CS	¥19,800	¥14,200
C-TRACE68 Ver.3.0(キャスト)	¥98,000	¥50,000
サイクロンEXPRESSα68	¥98,000	TEL下さい。
G68K Ver2 PRO	¥22,000	¥17,600
THE FILE PROFESSOR(ロゴシステム)	¥28,000	¥22,400
Gツール(ザインソフト)	¥28,000	¥22,400
たーみのる2(SPS)	¥17,800	¥14,200
マジックパレット(ミュージカルプラン)	¥19,800	¥15,800
Hyper word CZ-251BS	¥39,800	¥31,800

●ゲームソフト20%OFF OK.!!(一部ソフト除く)

周辺機器コーナー(送料¥500)

商品	定価	特価
Ⓐ CZ-8NSI	¥188,000	¥145,000
Ⓑ CZ-6VTI	¥69,800	¥54,000
Ⓒ CZ-6TU	¥33,100	¥25,000
Ⓓ BF-68PRO	¥19,800	¥15,500
Ⓔ CZ-6BEI	¥35,000	¥26,500
Ⓕ CZ-6BEIA	¥38,000	¥28,600
Ⓖ CZ-6BE2	¥79,800	¥60,000
Ⓗ CZ-6BE4	¥138,000	¥107,000
Ⓘ CZ-6BFI	¥49,800	¥38,200
Ⓙ CZ-6BPI	¥79,800	¥61,000
Ⓚ CZ-6BMI	¥26,800	¥20,300
Ⓛ CZ-6EBI	¥88,000	¥67,500
Ⓜ AN-S100	¥36,600	¥28,500
Ⓝ CZ-6SDI	¥44,800	¥35,000
Ⓞ CZ-8PC3	¥65,800	P&A超特価TEL下さい。
Ⓟ CZ-8PC4	¥99,800	P&A超特価TEL下さい。
Ⓠ CZ-8PG1	¥130,000	P&A超特価TEL下さい。
Ⓡ CZ-8PG2	¥160,000	P&A超特価TEL下さい。
Ⓢ CZ-8PK10	¥97,800	P&A超特価TEL下さい。
Ⓣ CZ-6PVI	¥198,000	¥153,000
Ⓤ IO-735X	¥248,000	¥190,000
Ⓥ CZ-8BSI	¥23,800	¥19,000
Ⓦ PIO-6BE1-A(I/O DATA)	¥25,000	¥18,000
Ⓧ PIO-6BE2-2M(I/O DATA)	¥50,000	¥36,500
Ⓨ PIO-6BE4-4M(I/O DATA)	¥88,000	¥64,300

X68000用ハードディスク(送料¥1,000)

アイテム
- ●HXD-040(40MB/23ms)……定価¥118,000▶特価¥88,000
- ●HXD-042(増設用)……定価¥128,000▶特価¥95,000

アイテック
- ●ITX-640(40MB/28ms)……定価¥158,000▶特価¥89,000
- ●ITX-680(80MB/20ms)……定価¥198,000▶特価¥116,000

プリンター(ケーブル・用紙付)限定5台 新品(送料¥1,000)

- ●CZ-8PC3(カラー漢字24ドット熱転写プリンター) 定価¥65,800……特価¥39,800
- ●CZ-8PK8(24ピン漢字プリンター136桁) 定価¥152,000……特価¥49,800
- ●CZ-8PC4 P&A特選!!(カラー漢字48ドット熱転写プリンター) 定価¥99,800……特価¥59,000

モデムコーナー (送料¥1,000)

商品	定価	特価
Ⓐ MD-24FS5(オムロン)	¥49,800	¥34,800
Ⓑ MD-24FS7(オムロン)	¥64,800	¥45,000
Ⓒ コムスター2424/4(NEC)	¥38,800	¥28,000
Ⓓ コムスター2424/5(NEC)	¥44,800	¥32,000

P&A特選パソコンラック (送料無料)移動自由(キャスター付)

中古パソコン(セットはモニター付)送料¥2,000

- ●X68000セット……▶¥180,000
- ●X68000 ACEセット……▶¥200,000
- ●X68000 ACE-HDセット……▶¥215,000
- ● EXPERTセット……▶¥230,000
- ● EXPERT-HDセット……▶¥265,000
- ● PROセット……▶¥250,000
- ●X68000PRO-HDセット……▶¥270,000
- ● EXPERTⅡセット……▶¥250,000
- ● EXPERTⅡ-HDセット……▶¥320,000
- ● PROⅡセット……▶¥240,000
- ● PROⅡ-HDセット……▶¥310,000

中古パソコンはP&Aにおまかせ!!

その場で高価現金買取り・高価下取りOK!!

■まずはお電話下さい。
03-651-1884 FAX:03-651-0141

■下取り・買取りでお急ぎの方、直接当社に来店、または、宅急便にてお送り下さい。

●下取りの場合……価格は常に変動していますので査定額をお電話で確認して下さい。(差額は、P&A超低金利クレジットをご利用下さい。)

●買取りの場合……現品が着き次第、2日以内に買取り金額を連絡し、振込み、又は書留でお送り致します。

●近郊の方は、P&A本店まで、直接お持ち下さい。即金にて、¥1,000,000までお支払い致します。

《便利な超低金利クレジットをご利用下さい》

●月々¥1,000円からOK.!! ●ボーナス払いOK(夏冬10回までOK)
●支払い回数 1回～84回 ●お支払いは、8ヶ月先からでもOK.!!

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)

〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 (株)ピー・アンド・エー

〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回～84回払いまで出来ます。但し、1回のお支払い額は¥1000円以上。

超低金利クレジット率

回数	3	6	10	12	18	24	36	48	60	72	84
手数料	3.0	4.0	5.5	5.5	10.0	11.5	16.0	21.0	27.0	35.0	42.0

●マイコン
●ビデオ
●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目1番地19号
☎03-651-0148(代) FAX.03-651-0141

営業時間
平日:AM10:00～PM7:00
日祭:AM10:00～PM6:00

●現金書留及び銀行振込でお申し込みの方は、上記商品の料金に3%加算の上でお申し込み下さい。詳しくは、お電話でお問い合せ下さい。

オー! ダイナ
10月創刊
定価580円
DynaBook/J-3100シリーズで使えるビジネスアプリケーション情報を中心に、ハードウェア・周辺機器からOS環境まで、幅広いジャンルにわたるサポートを実現。最新情報はもとより、初心者から上級者までそれぞれユーザー自身がレベルに合わせて読み進めるように編集。また、解説・入門記事などはすべて付録の3.5インチディスクの内容と連動し、新しいスタイルのコンピュータ情報誌として、ユーザの皆様のニーズに対応。
ダイナブック
Oh! Dyna
SOFT BANK
ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13
☎03(5488)1360:営業部

東芝DynaBooK/J-3100シリーズ
アプリケーション活用誌「オー！ダイナ」
Oh! Dyna
急接近。
3.5インチディスク
付録
特集
ハードウェアやジャンル別アプリケーションの活用
DynaBook/J-3100ハードの活用、アプリケーション、周辺機器、OS、通信、ファイルコンバート、付録ディスクのアプリケーションの紹介など
SPECIAL REPORT
IBM PCより移植が決定された日本語版主要アプリケーション・OSなどの製品レポートまたは、GUIやネットワーク関連の応用と実践
NEW PRODUCTS
最新ハードウェアおよびアプリケーションのレビュー
入門・活用
DynaBookの活用やWindows、ネットワークなどをテーマに初級、中級、上級レベルの3段階で構成
連載
エッセイやコラム、企業ユーザーレポート
DynaBook
J-3100GS

ツクモぱそこん秋場所

掲載商品2万円以上**送料無料!!**

ツクモ通販受注センターフリーダイヤル
0120(377)999
商品のお問い合せは各店又は通販部☎03(251)9911へ

冬のボーナス一括払受付中！詳しくはお問い合せ下さい。

NEC・エプソン・東芝・富士通・各メーカー取り扱っております

横綱 X68000シリーズ

PROⅡ CZ653C 定価¥285,000 / CZ663C 定価¥395,000
●次世代のインテリジェンス、SX-WINDOW搭載●知的ニュースタンダードフォルム●BIOSの改良によりハイスピード処理を実現●2Mバイトの大容量メモリを標準装備●拡張I/Oポート4スロット標準装備

EXPERTⅡ CZ603C 定価¥338,000 / CZ613C 定価¥448,000
●次世代のインテリジェンス、SX-WINDOW搭載●象徴のフォルム、マンハッタンシェイプ●BIOSの改良によりハイスピード処理を実現●3Mバイトの大容量メモリを標準装備

SUPER HD CZ623C 定価¥498,000
●次世代のインテリジェンス、SX-WINDOW搭載●「チタン」カラーのクォリティブラック●80MB SCSIハードディスク搭載●世界標準SCSIインターフェース標準装備●BIOSの改良によりハイスピード処理を実現●3Mバイトの大容量メモリを標準装備

ツクモ特価販売中!!

大関

光磁気ディスクユニット

シャープ
CZ-6M01 予約受付中！
SCSIボード
CZ-6BS1 予約受付中！

ハードディスク

アイテック
IT X640
定価¥158,000 特価¥89,800

IT X680
定価¥198,000
特価¥118,000
(カラー：ブラックとグレー)

一流メーカー
40MB HDD
限定特価
¥59,800

「MIDIサウンドライブ in オータム」

♬大好評のMIDIライブショーの第2弾です！

開催日　9月22日(土)・23日(日)　午後3回
(12:30/1:30/2:30)30分ずつ。

場所　九十九電機㈱7号店 1階店頭にて、X68000とMIDI楽器を使った楽しい演奏♬クイズもあって景品が当る！是非お立ち寄り下さい。

♪♪♪ X68000用MIDI ♬♬♬

Aセット		Bセット	
CM-32L	¥69,000	CM-64	¥129,000
SX-68M	¥19,800	SX-68M	¥19,800
Musicstudio Mu-1	¥19,800	Musicstudio Mu-1	¥19,800
合計定価	¥108,600	合計定価	¥168,600
ツクモ特価	¥91,800	ツクモ特価	¥144,000
	(消費税別途¥2,754)		(消費税別途¥4,320)
クレジット例(税込)	月々¥5,830×18回払	クレジット例(税込)	月々¥7,107×24回払

★Musicstudio PRO-68K V1.1又は、Music PRO68K(MIDI)のソフトの場合には¥8,000プラスになります。

X68000用メモリーボード

1MB増設用メモリーボード ACE&PROシリーズ内蔵用
ツクモ特価 ¥19,800 (消費税別途¥594)

2MB拡張RAMボード……………ツクモ特価¥39,800 (消費税別途¥1,194)

4MB拡張RAMボード……………ツクモ特価¥69,800 (消費税別途¥2,094)

※2MBと4MBは全シリーズ対応。拡張スロット用。

アートツール

一流メーカー
イメージスキャナ
最大A4サイズの絵や写真をフルカラー読み取り
大特価目玉品 ¥128,000

カラーイメージユニット
512ドット、65,536色の鮮やかな映像入力。
CZ-6VT1
定価¥69,800

ビデオボード
CZ-6BV1 定価¥21,000

ツクモ特価販売中！

通信モデム＆ソフト

一流メーカー
2400ボークラス4 定価¥54,800
ツクモ特価¥28,000 (消費税別途¥840)

た～みのる2 定価¥17,800
ツクモ特価¥15,000 (消費税別途¥450)

Communication PRO-68K
ツクモ特価¥19,800 (消費税別途¥594)

お勧めソフト番付 (税・送料別)

西 ツール&ビジネスソフト			東 ホビーソフト		
関脇	SX-WINDOW	¥6,800	関脇	Sim City	¥9,800
小結	C compiler PRO-68K Ver.2.0	¥44,800	小結	ポピュラス	¥9,800
前1	CYBERNOTE PRO-68K	¥19,800	前1	ダンジョンマスター	¥9,800
前2	Hyper WORD	¥39,800	前2	サンダーブレード	¥9,500
前3	マジックパレット	特¥16,830	前3	ワンダラースフロムイース	¥8,700
前4	サイクロン Express α 68	特¥83,300	前4	ワールドコート	¥8,800
前5	デジタルクラフト	特¥33,830	前5	ギャラガ 88	¥8,200
前6	Z's STAFF PRO-68K Ver.2.0	特¥49,300	前6	クオース	¥6,800

大好評入会者募集！

ツクモグローバルカード

国内・外で活躍！
使って便利、持ってて安心！ツクモグローバルカードはジャックス・VISA、セントラル・マスターとの提携カードです。ツクモ各店でのお買物がらくらくできるうえに、国内はもとより海外でのショッピングもOK！しかも18歳以上なら学生でもOK！

18才以上なら学生でもOK！

冬のボーナス一括払・金利手数料無！受付中!!
お申し込みは(03)251-9898又は各店で

●うぃっぷす2　●GEM DROP

パソケットなど，同人市場で流通しているプログラムの紹介コーナーです。単に同人ソフトという枠でなくユーザーの作ったプログラム紹介して，個人で作った大規模なプログラム，ネット上のシェアウェアなど，広く紹介していきたいと思います。

●

同人ソフト紹介の第1弾としてX68000を中心とした同人サークルEXIZの作品を2本紹介しよう。

●うぃっぷす2（X68000用）

画面中の鍵を拾って鏡の扉を抜ければ1面クリア。障害としては，丸いパックマン風の敵キャラが追いかけてくる。

一見するとアクションゲーム風に見えるが，主人公キャラクタの女の子になぜかオプション（?）が3つついているというのがゲームにパズル性を与えている。オプションの間隔を変えることで敵キャラをうまく停止させたり，すり抜けたりでき，最低限の危険性でことを運ぶのがゲームのポイントとなる。あまりに安全策をとっているとタイムオーバーになることもある。

起動してゲームを始めると，前回までの最高面までの面セレクトモードになるなどの気配りもよい。基本システム以外にも面データを集めたディスクがあり，末長く楽しめる。

通販価格　基本システム　1,000円
　　　　　追加データ1～3　各700円

●GEM DROP（X68000用）

カラムスのようなハットリスのようなリアルタイムパズルゲーム。画面上方から落ちてくる宝石を一定のパターンで消していくというテトリス以来ありがちなパターンではある。

初心者モードでは宝石の色か形が3つ揃えば画面から消えるようになっている。上級モードでは消える条件がシビアになる。カラムスなどと違う点は落ちる場所を指定する際にブロックではなく，落ちる場所のほうを動かすことだろう。左右がつながっているので，ときとして予想以上のブロックが消えることがある。

グラフィック，ミュージック，プログラムなど総じて完成度は高い。宝石の落ちる場所は制限されていてもカクカク感はない。この点では一部の市販ソフトにも優る。この手のブロックゲームはやり込めばそれなりに面白いので，とりあえずおすすめ。

通販価格　1,500円

うぃっぷす2の画面。キラキラした鏡が美しい。しかし，ただ走り回ればいい面とオプションを駆使しなければならない面との格差が大きいんじゃないかい？　という気もしないではない。

●入手方法

今回紹介したソフトの購入希望者は以下のものを用意してほしい。

1)　申し込み内容を書いた手紙
2)　料金分の無記名の郵便小為替
3)　返送先を書いた宛名シール

1）は当たり前。自分のほしいものを明記し，通販希望の旨を伝えること。

2）は郵便局で買ってくる。名前を書く欄があるが，なにも書いてはいけない。料金はソフト代＋送料分となる（送料はディスク何枚でも一律300円）。

3）は発送をスムーズにし，郵便事故を防止するためにも必ずつけること。文房具屋で無地のタックシールを買っておくとよい。どうしようもない場合は，適当な大きさの紙に自分の住所氏名を書いて同封する。トラブル防止のため，宛名シール以外のところにも自分の住所を書いておくこと。

申し込みは以下の宛先に封書で。

〒223 神奈川県横浜市港北区箕輪町
2-18-27 田口方
同人サークルEXIZ

＊

さて，このコーナーでは引き続き同人ソフト，その他の紹介を行っていく予定。誌上での紹介を希望するサークルは，Oh!X編集室「THE USER'S WORKS」係まで連絡を。ただし，スペースの都合，およびソフトの内容によっては掲載できないこともあるのであらかじめ了承してほしい。

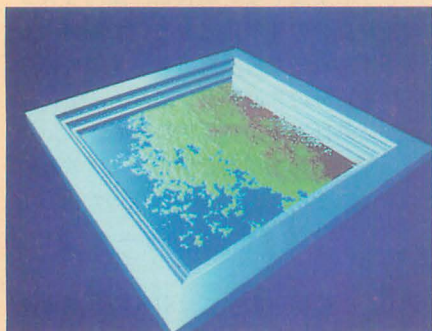
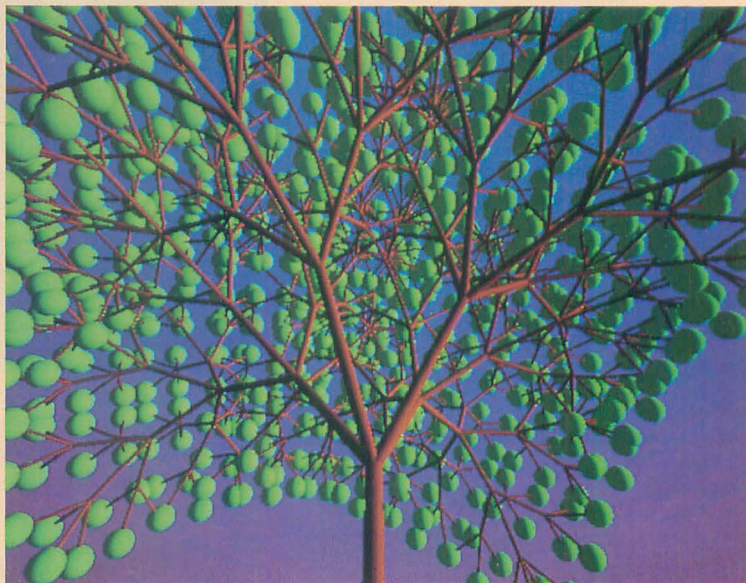
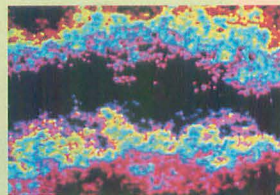
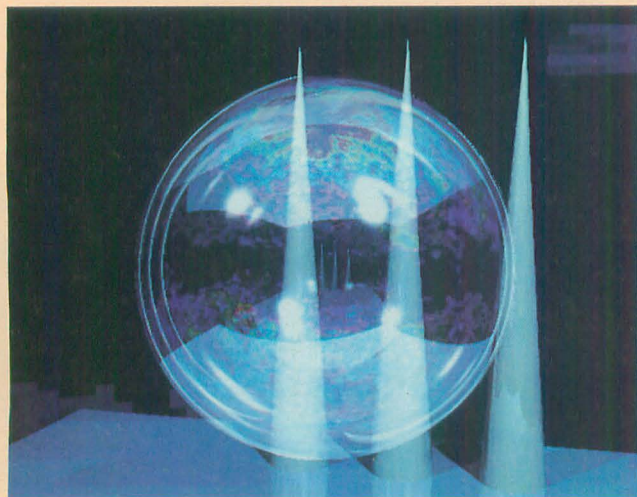
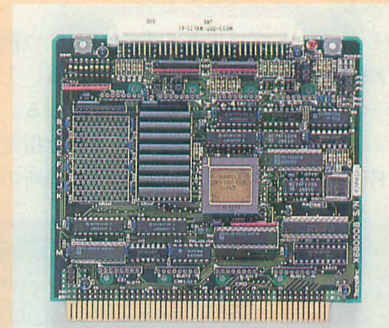
面が進むと落ちてくる宝石の数も増え，速度も速くなってくる。入門モードのほうがルールとして面白いのだが，プレイヤーの理解を超えたところでブロックが消えていくのは問題かもしれない。

C-TRACE68 ver.3.0 & C-TRACE68 TP

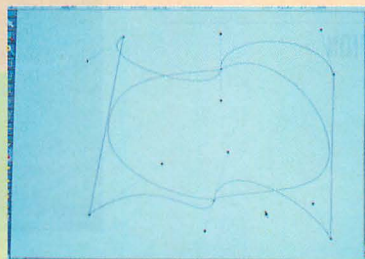
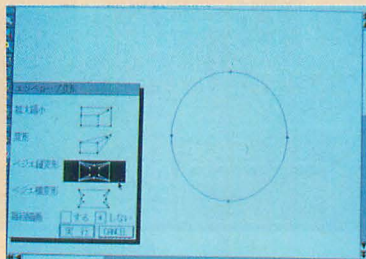
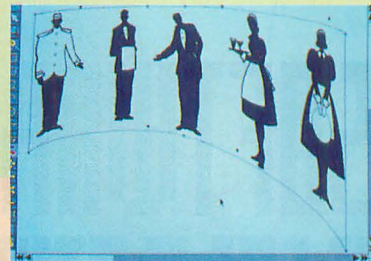
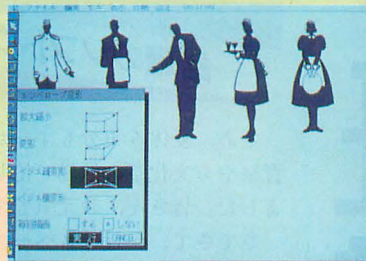
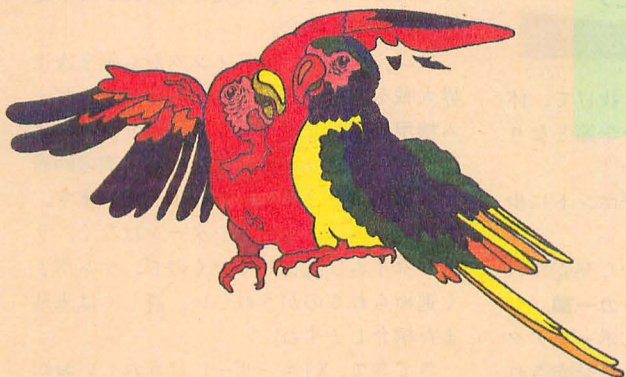
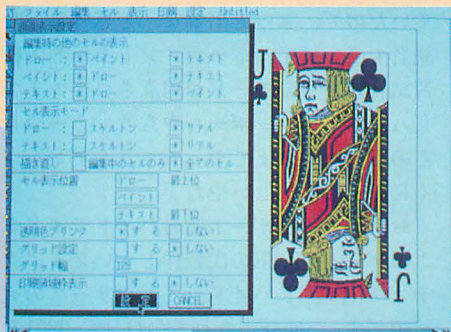
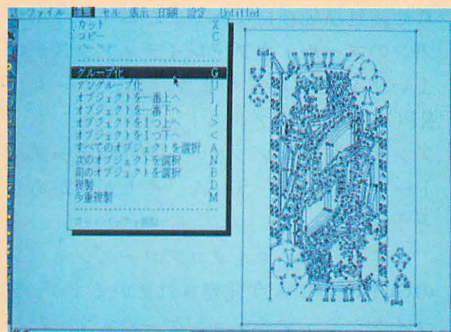
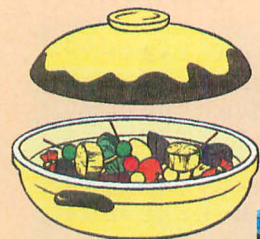
C-TRACE68 ver.3.0＋トランスピュータによるサンプル画像。従来は物体数の2乗に比例した計算時間がかかっていたため，レイトレーシングで複雑な物体を描くにはそれなりの覚悟が必要だった。そこで現れたのが交点計算を軽減するボクセル分割だ。ボクセル分割は物体数が増えるほど効果が表れてくる。

さらにトランスピュータを搭載すれば劇的に計算速度が向上する。どれくらい速いかというと,「ここで挙げた程度のレイトレーシングなら手軽に作れる」くらいには速い。

いちばん大きな写真の再帰樹木の場合，1本の枝が4つに分割されて……というのを5段階再帰した結果，テキストデータに換算して32000行の形状データとなった。これがモデリング(データはC言語による自動生成)，テスト計算，本計算まで含めてひと晩がかりだ(データ生成プログラムの作成時間含む)。これを速いと見るか？　まだ遅いと見るか？

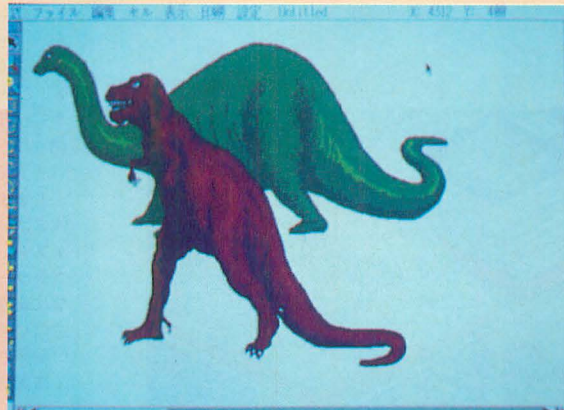
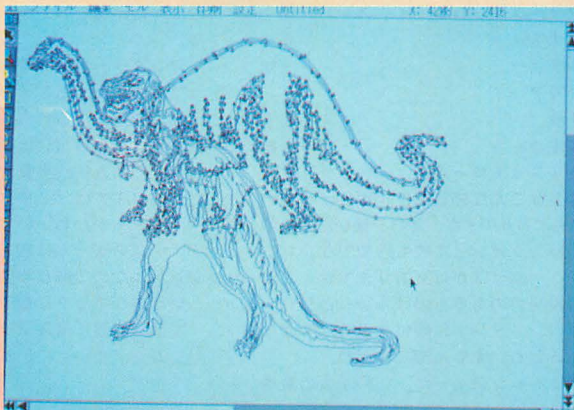
透過率マッピングで描いたステゴちゃん，バンプマッピングの時計文字盤，まわりの中に青空のカラーマッピング（これもフラクタルで作った）を張り付けたもの。マッピングにもいろいろある。

同系列のフラクタルでカラーマッピングとバンプマッピングを作成して張り付けた地図。気のせいか，ポピュラスを思い出させる。

シャボン球の背景には上のような図形がマッピングされている。映り込みを利用したトリッキーな画像。マッピングにはフラクタルがよく似合うようだ。

これがトランスピュータボード。ボード上に並んだ8本のRAM。これで4Mバイト。そう，4Mビットチップを使っているわけだ。隣に8本並べると8Mバイトに増設でき，さらに子ボードをこの上に4枚装着してトランスピュータ5台の並列計算が可能。

ポップアートツールCANVAS PRO-68K

シャープから近く新しいグラフィックアートツールが登場する。先月もちらっとお知らせしたように，X68000では初めてのドロー系グラフィックツールだ。開発も着々進んでいるようで，編集部でもその開発途中バージョンを見せていただいた。まだ，詳しいレビュー記事が書ける状況ではないが，写真をご覧いただければ，どういうコンセプトのツールかなんとなくおわかりいただけるだろう。商品名はCANVAS PRO-68Kに決まったとか。ともかく期待の一作だ。完成を待ちたい。

CANVAS PRO-68Kでは，ベジェ曲線と呼ばれるドローラインをもとにいろいろな図形を作成する。画面は円のコントロール情報をマウスでいじって変形させたところ。

作成したオブジェクトは自由に変形することができる。画面は複数のドローオブジェクトに対してベジェ縦変形というエンベロープ化を行った例。

ドロー型ツールのメリットは，少ないメモリで絵を表現でき，拡大しても滑らかなラインが崩れない。特にプリントアウトすると違いがはっきりする。なお画面はドローオブジェクトをスケルトンモードで編集したときと，リアルモードで表示したときだ。

複雑なドローオブジェクトをいくつも組み合わせて作成する。ドローオブジェクトは表示順位を変えたり，グループ化して編集したりもできる。また，作成したオブジェクトをペイントセルにコピーして，ドローセルと重ね合わせ表示もできる。

THE SOFTOUCH

SOFTWARE INFORMATION

熱血高校ドッジボール部サッカー編
人気ゲーム熱血高校シリーズ最新作。またまた登場のくにおくん，今度はサッカーに挑戦だ。

今月も新しい情報は少なかった……。期待して待っていた読者の方々，ごめんなさいね。もうちょっとすると，年末年始に出されるソフトがどばどば発表されると思うから，ね。もうちょっと，だよ。

話題のソフトウェア

さあ，夏休み気分もすっかり抜けて，体育祭やら文化祭の準備で忙しい季節となりました。皆さん，お元気ですか？

さてさて，今月はスペースがホントに少ない！　さっさとやらねばね。

今月のトップは，シャープから発売される**熱血高校ドッジボール部サッカー編**。お馴染みくにおくんが活躍するスポーツアクションゲームです。ファミコンで発売されていたので，きっと出るな，と思っていた読者も多いはず。トーナメントモードと練習試合の2つのモードが用意されています。しっかし，ドッジボールやらサッカーやら，そのうえケンカまでやっちゃうんだから，くにおくんも忙しいやねぇ。

お次は，ブラザー工業の**ハイドライド3**。このゲーム，T&Eの代表作ともいえるアクティブRPGで，ハイドライドシリーズの完結編です。今回発売されるのは，PC-9801などで発売されていた“ハイドライド3SV”の移植版，X68000の特性を生かした仕上がりになっているそう。発売は10月13日の予定です。お楽しみに。

はい，次。アートディンクから第2次世界大戦を題材にしたシミュレーションゲーム**機甲師団**が発売されます。取り込みを駆使したヨーロッパの地図やセピア調の画面が，雰囲気をうまくかもし出しています。このゲーム，ヘックスやターンはなく，リアルタイムに進行していくので，テンポよく進められるのがうれしい。詳しくは来月また紹介しますね。

さてさて，X1ユーザーにはうれしいお知らせ。延び延びになっていたポニーキャニオンの**プール・オブ・レイディアンス**がいよいよ発売されるもよう。このゲームはRPGで，アドヴァンスト・ダンジョンズ・アンド・ドラゴンズという海外のテーブルトークの移植版。うーん，やりがいがありそうですね。

んでもって，マイクロプローズジャパンの**GUNSHIP**。もう発売されていますが，フライトシミュレーションゲームです。なかなか実際に操縦している気分にひたれてい

期待票で大物2作がランクインだ！

順位	タイトル	前回
1	ポピュラス(含プロミストランド)	1
2	スーパーハングオン	7↑
3	三国志II	−↑
4	ダンジョンマスター	2↓
5	ワンダラーズ・フロム・イース	10↑
6	ギャラガ'88	6
7	シムシティー	−初
8	ソーサリアン	8
9	グラナダ	3↓
10	ワールドコート	−初
	トンネルズ&トロールズ	9↓

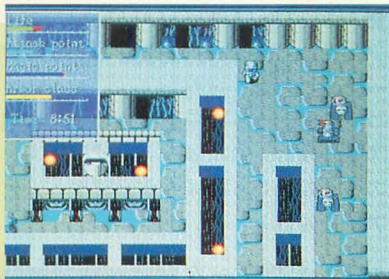
とまあ今月はこんな感じです。ポピュラスの強さはいつもどおりですが，それを追いかけるのがスーパーハングオンってのも妙ですな。ロングセラーのゲームはほかにもありますが，こんなに得票に瞬発力のあるものはちょっとない。逆にいえばコンスタントに順位をキープできない弱味があるともいえるんですけどね。

三国志IIとシムシティーが，X68000版の登場前だというのにチャートに入ってきました。どちらもトップを争える強豪だけに，ポピュラスにとっては不気味な存在。大作と目されながら，いまだトップを取っていない三国志II。最近人気の海外ソフトの大御所だとだれしも認めるシムシティー。ポピュラスからトップの座を奪うのは，たぶんこの2作に絞られるんじゃないかな。

ワンダラーズ・フロム・イースは5位まで戻ってきました。上と下を見ると「うーん，やっぱりこの辺になるだろうなぁ」という感じ。イースI・IIの人気には届かないみたいです。

今月の新人は，実質ワールドコート1本でした。ハガキでは「2〜4人の対戦がいい」「キャラクターがかわいい」などの感想がでています。テニスゲームは敵も少ないし，人気もあるジャンルなので伸びる要素は十分ですが，要2Mというのがネックになるかも。

では，次回「海外ソフト対決──ポピュラス対シムシティー」に，イナズマキィーック!!

（浦）

ハイドライド3

機甲師団

プール・オブ・レイディアンス

いもんです。空中にいる感じがなんとなくしてくるから不思議なもんですね。

それでは，いきなりタケルの情報です。ブラザー工業ではただいまX1用に**セレクテッドソーサリアン5とヴァーチュアナイト**，X68000用には**FLY**を発売中。セレクテッドソーサリアン5は，ソーサリアンシリーズのシナリオ集。もう何もいうことはないでしょう。ヴァーチュアナイトはLOGINソフトコンテストの入賞作品で，横スクロールタイプのRPGです。FLYも同じくLOGINソフトコンテストの入賞作品で，こちらは磁力を持つ球をスピード調節しながらコースを回るゲーム。どちらもお手頃価格2,000円なのでサクサクッと遊ぶにはよさそうです。しっかし，X1でRPGばっかりになっちゃったなあ。

最後に，光栄からは**三国志II**のX68000版が発売中，データウエストからは**Misty5**が今月末に発売される予定。じゃ，また。

GUNSHIP

Oh!X読者代表，ポピュラス大会で優勝!(した人に負けちゃった)

8月号でお知らせしたポピュラス大会が予定どおり，7つのパソコン誌の読者代表，およびイマジニア代表によってとり行われました。

まずは，Oh!Xの代表の紹介から。われらがOh!Xの代表は熾烈な全国大会の末(本当かなあ)，横浜市にお住まいの大学院生，田中啓介君(24)に決定しました。彼は大学の研究室で，日夜(?)対戦に励んでいたそうで，かなりの腕前。ハガキの文面によると「この大会は私のためにあるようなものです」とのこと。うーん，これで1回戦で負けたりしたら，ただじゃあ……。

さて，いよいよ決戦当日の8月18日(土)。THE COMPUTER館6階の大会会場に足を踏み入れると，そこにはズラリと並んだPC-9801が……。しかし，決戦場所の隣にはゲームコーナーなる，自由に「ポピュラス」，「シムシティー」を遊べる場所が用意されていて，そこにはX68000やFM TOWNSもあったぞ。よかったね（?）

さて，われらが田中君の様子はというと，これが結構リラックスしている模様。一方，ほかの人たちはどうかなと見回してみると……，「やっぱり」，予想どおりLOGIN代表はなにか衣装を用意している。さすがはLOGIN。

それともうひとつびっくりしたのが，コンプティーク代表。なんと女の子（人?）じゃあないか。う〜ん，編集者の趣味だろうか。うちはパソコンギャルでも出せばよかったかな。

そして気になるのが，イマジニア代表。「やはり，主催者の面目にかけて，年棒1億円ぐらいで外人助っ人ゲーマーでも引き連れて来るんだろうか」とかと，いろいろ考えていたのだが……。実際にイマジニアの人から紹介されたのは，これまた女の人（イマジニアの社員だそう)。いいなあ，むさくるしい男ばかりの中に花が2つ添えられたようで。しかし，これはぬいぐるみで中に作者のピーター・モリニュー氏が入っているのかもしれないから警戒しようっと。

いよいよ，1時。戦いの始まりである。まずは，くじ引きで対戦相手の決定。うちの相手はOh!PCに。いきなり，内部抗争か。いやな，雲行き。まあ，しょうがない，死んでもらおう。

第1回戦

Oh!PCの代表はなかなか手強く，混戦模様。それに加えて，機械の調子が悪いのか，ケーブルが悪いのか，しょっぱなからハングアップの連続。結局最後までハングが続き，勝敗も判定でうちの勝ちということになった。人口などを見ても最後にはほとんどこっちの勝ちは決まっていたとはいえ，ちょっとかわいそうだったOh!PC代表でした。初めのうちははしゃいでいた田中君にも壮絶な戦いに疲労の色が少し見える。

第2回戦

今度の対戦相手はイマジニア代表に圧勝したマイコンBASICMagazineの代表。1回戦の戦いぶりを見たけれど，どれくらいの強さかはわからなかった。しかし，いざ試合が始まってみると，結構あっさり勝ってしまった。

意外と早く試合が終わったので，もうひとつのほうの対戦であるLOGIN対コンプティークに目を移してみると……。LOGIN代表もさることながら，コンプティーク代表の女の人もなかなか強い。これなら代表に選ばれたのも納得できる，というぐらいなのでした。認識の甘さに反省することしきり。対戦の結果のほうはというと激戦の末，LOGIN代表が見事勝利をおさめ，決勝戦へと駒を進めたのでした。

いよいよ決勝戦

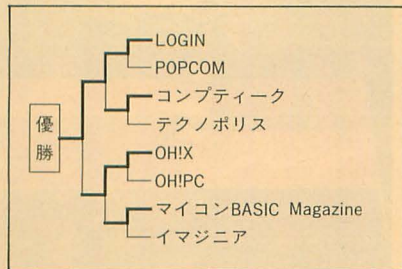
さて，いよいよLOGIN代表との優勝争い。このLOGIN代表，実は田中君の研究室でいつも対戦しているライバルだそうで，因縁の対決というか，世間は狭いというか。

田中君はなんとOh!Xをマウスパッドに使っているぞ! ありがたいやら，もったいないやら(写真左)。LOGIN代表は原始人?(写真下)

両者の対決は実際の戦いぶりもさることながら，舌戦のほうがこれまたすごい。ほとんど，周りのことなど気にもせず，研究室での対決を地でいってるなという感じでした。口では田中君のほうが勝っているようでしたが，戦いのほうは田中君がだんだん不利な状況に追い込まれていってしまい，ゲームエンド。

というわけで残念ながら，田中君は結局準優勝に終わったのですが，がんばってくれました。皆さん，彼の健闘をたたえてあげてください。

会場で気になったのが，対戦には目もくれずに黙々とゲームコーナーで遊んでいる人たち。みんな，あんまり人の対戦には興味がないのかな。あと，見に来ていたOh!Xの関係者の多いこと。みんな，ポピュラス好きだから。しかし，少しうちばかり騒ぎすぎたという感もあります。みなさん，どうもご迷惑をかけました。

しかし，まあ並み居るゲーム雑誌の中で（ゲーム雑誌でない）Oh!Xの読者代表が準優勝というのは，かなり名誉なことではないでしょうか，えらいえらい，と日記には書いておこう。というわけでポピュラス大会のレポート終わり（あ，ちなみに後日来日したピーター・モリニュー氏はそんなに強くなかったそうです)。

THE SOFTOUCH

●プロテニスワールドコート

友達とやるのが○のテニスゲーム

Yamada Junji
山田純二
Kageyama Hiroaki
影山裕昭

X68000にひさびさに登場した正統派スポーツゲーム。もともとはナムコのアーケード版ですが，PCエンジンにも移植されているので知っている人も多いでしょう。対コンピュータでの必殺技も公開しちゃいます。

X68000用　5"2HD版2枚組　8,800円(税別)
SPS　☎0245(45)5777

AM1：20チャッチャカチャ～。

皆さん今晩は，「今日も元気でスポーツどん！」の時間がやってきました。今日は新発売の「ワールドコート」の対戦を生中継でお送りしていきます。それでは，影山さん，山田さん，どうぞ！

からんころ～ん♪

影＆山：こんばんは～。

アナウンサー(以下ア)：おはようございます。ごめんなさいね2人とも，わざわざこんな時間に呼び出してしまって。

山：まあ，暇でしたから。

影：僕は，今日は山田君の泣き顔を見にきただけですから。

ア：おやおや，影山さんやけに自信ありそうですね。

影：もう以前の僕ではありませんからね。

ア：山田さんはどうですか？

山：やってみればわかりますよ。

ア：そりゃそうだ，というところで始めましょうか。

影：それじゃあ，僕はチナチロパで。

山：僕は，サボテン。コートはロンドン，1セットマッチでいいですね。

影：いいよ（ポチ)。

X68000：ラブオール（0－0)。

影：まずは，僕からのサービスゲームですね。いきま～す（パシッ)。

山：そんなへなちょこサーブ，俺には通用せん（すかっ)！

X68000：フィフティーンラブ（15－0)。

山：おやあ？

影：はっはっは～の，もひとつ“は”。このゲーム，軽くキープさせてもらいますかサーブ（パシッ)！

山：そうは馬屋のけつレシーブ（パシッ)。

影：素直なレシーブ命取り，クロスあたたたたっく！

X68000：サーティーンラブ（30－0)。

影：次いくぞ！　ていっ（パシッ)。

山：あじゃぱっぱ～(スカッ)。

コーチ（以下コ)：山田！　なにやってんだ，俺が教えたことを忘れたのか。

山：す，すいませんコーチ。

ア：おっと，いきなりわいて出てきたUコーチ，「俺が教えたこと」とは？

コ：必殺技です。

ア：はあ，そうですか。

影：ふん，なにが必殺技だ。次のゲームも軽くブレイクさせてもらいましょうか。

山：ふっふっふ，聞いて驚け見てびっくり。必殺（これを使えば，ハムスライスでコンドルに勝てるかもしれない)天井サ～ブ(ポヨヨン)。

影：なんじゃこりゃ（スカッ)。

ア：おおっ確かに，意表をつくサーブ。コーチ，これはいったい……。

コ：まず，ボールを上げてすぐに空振りをする。次に，Bボタンでロブ。すると，今のような山なりのサーブが打てるんです。

ア：なるほど。

さて，最終ゲーム

ア：ポイント，2（影)－5（山)，となりいよいよ最終ゲームか，すでにカウントは30－40。調子を取り戻した山田に影山，苦しいゲーム展開となってしまいました。

影：くっそ～，俺の血と汗と涙の特訓はなんだったんだ。

山：無駄だったんじゃない。

影：ぬうううう（パシッ)。

山：これで，ひと思いに決めてやるレシーブ（バシッ)！

ア：あーっと，前に出ていた影山の逆をついた痛烈なレシーブ，これで決まったか。

影：飛んでくれー！（バシッ)

X68000：デュース。

ア：なんとかくらいついて，デュースに持ち込んだ影山。しかし，このあとすぐにダブルフォールト，アドバンテージ山田だが，両者必死のラリーを続けています。

影：うおっ，とえっ，だ～！

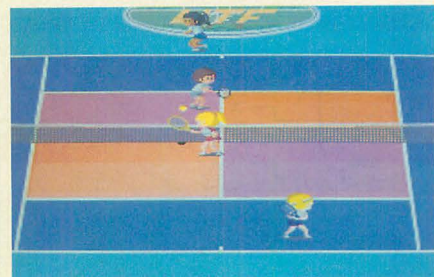
山：このっ，このっ，うりゃ！　げっ，しまったー。

影：（にやり）はっはっは，ラッキーボールをありがとう，まだ負けるわけにはいかないぞスマーッシュ！

X68000：アウト。ゲームセットアンドマッチ。

影：うそだ～。1ドットライン上にのってるじゃないか。

ア：あ，影山さんディスプレイを揺すらないで。コーチこれはどういうことなんでしょうか。私には確かにライン上にボールがのっているように思われるんですが。

コ：そうですね，これはばっかりはどうしようもないんじゃないですか。ときどきこういったことが起きるようですね。

ハデなコートで4人でルンルン♪

ア：そうですか，とりあえず山田さんの勝ち。山田さんどうでした，この試合？
山：影山さん，まだまだ修業が足りないようですね。
影：うそだ，うそだ……。
ア：おやおや，影山さんいつまでもめげていないで，今度は気分を変えてダブルスでもやってみませんか。
影：うう，いいですけど。山田君，ダブルスのあともうひと勝負やりましょう。
山：どっちにしても僕が勝ちますけどね。
ア：すっかり熱くなってしまった2人。今度はダブルスですから，協力してがんばってくださいね。
影&山：は〜い。

てなことでダブルス

山：どうせなら男女混合にしてみない？
影：いいねぇ。じゃ僕コンドル（ポチ）。
山：じゃあ僕はサボテンにします（ポチ）。
影：コンピュータはマケロウとステッピイでいいよね(ポチポチ)。コートはカラフルなニューヨークにしてゲームスタート。
X68000：チロリン。ラブオール（0−0）。
影：僕のサービスからね（ポンポンポン）。ステッピイごときに俺の稲妻サーブが取れるかな？　それっ！（バシッ）。あれー？
X68000：フォールト。
ア：わずかにラインを越えてましたね。
山：次はしっかりしてくださいよ。
影：ああ（バシッ）。げっ，やばい。
X68000：ダブルフォールト。ラブフィフティーン（0−15）。
ア：あれあれ。いきなりダブルフォールトですか。
山：ちょっとちょっと，真面目にやってるんですか。わかった。僕に負けて動揺してるんでしょ。
影：ま，まさか。これは愛敬さ。これくらいサービスしなきゃ。ま，ハンデかな。
山：そのわりには冷や汗かいてますけど。
影：そ，そんなことないさっ。
ア：さあ影山，気を取り直して再びサービスです。
影：どりゃあ（バシッ）。
ア：気合いの入った一発。さすがは男子プロ，球の速さが女子プロとは段違いですね。しかし，マケロウも負けてはいません。体勢を崩しながらも球を返しています。
山：チャーンス（バシッ）。
ア：前衛の山田，マケロウの返した球をすかさずボレー。これはうまい。しかしステッピイ，飛びついてこの球を拾ったぁ。

アナウンサーのおねーさんもグッドですよ

セレクトモード。硬派と軟派，どっちを選ぶ？

山：この生意気な！(スカッ)　やべっ，影山さんお願い。
影：了〜解（スパコーン)。
ア：後衛の影山も前衛の動きをよく見ていましたね。先ほどの敗戦の影響はもうないようです。前衛のステッピイは抜かれましたが，またもや後衛のマケロウが負けずに球を拾っています。
山：今度こそっ。ちくしょー。それっ。しつこいぞ！
ア：今度は前衛同士の激しいボレー合戦。どちらが勝つのか？　おっと，ステッピイがこれに耐え切れずロブを上げてしまいました。これは山田のチャンスボールです。
山：このこわっぱが。チョコ・ザイ・ナァー（バシッ)。しまったアウトだっ。
影：うまいっ。
X68000：フィフティーンオール（15-15)
山：？　完全にラインの外側だったのに。
コ：ダブルスになるとサイドラインが外側になるんです。
山：あっそうか。
ア：今のは見事なスマッシュでしたね。
山：あ，どうもどうも。
コ：それはそうと，読者から2人だけで楽しむなとクレームがつきそうなのでゲームの解説をしてください。
山：Uコーチから教育的指導を受けてしまいましたね。じゃゲーム中断（ESCポチ)。
影：ダブルスは女の子と和気あいあいと遊ぶもんだね。男同士でやるなら絶対シングルス。やっぱり人間が相手だと数倍面白い。
山：噂じゃ4人で遊べるモードがあるそうですよ。
影：その噂は本当らしい。なぜマニュアルに明記しなかったのか，疑問だけど。
山：やっぱりX68000にはジョイスティック端子が4つ欲しい。あとライン際の曖昧な判定がゲームを面白くしてるのも事実。
影：そうだよね。あの判定でゲームの流れがガラッと変わるし。泣きたいときもあれば笑っちゃうときもあるよね。
山：遊んでいてキャラクターに感情移入しちゃうよね。ナムコにこのテのスポーツゲームを作らせたら業界一だと思う。でも2MBないと遊べないのは辛いよね。てなところで，ぼちぼちゲーム再開といきますか。
影：いや，ダブルスはもうやめてシングルスにしよう。
山：でもゲームの途中ですよ。最後までやらないとタイトルに戻りませんよ。
影：あまーい。この秘孔インタラプトをつけば……。
一同：おおーっっ。
影：さっきの負けはマニュアルを読んでいなかったからさ。それに選んだキャラクターがいけなかった。僕はサボテンじゃないと駄目なんだよ。
山：そういう言い訳の応酬も対戦ならではの面白さですよね。
一同：ワッハッハッハ。
ア：これからシングルスのリターンマッチが行われるところですが，残り時間がわずかになってしまいました。誠に申し訳ありませんが，そろそろお別れしなければなりません。最後にひと言ずつどうぞ。
影：サボテンが1番。
コ：ハムスライスでしょ。
山：チョコ・ザイ・ナァ〜。
ア：それではみなさん，さようなら〜。

前方不注意，要2M！

ゲーム性，完成度ともに，「う〜ん，よしよし，う〜ん」なのですが，どうにかならなかったんですか，要2MB。初めてパッケージを見たときには理解に苦しみましたが，ゲームを始めた途端「はあ」というためいきとともに，すべてを悟ってしまった。確かに，あれだけのサンプリングデータとキャラクタデータを使えば，1MBでは収まり切れないかもしれません。それなら，1MBユーザーの場合はサンプリングデータをカットするなり，なんらかの対処をしてほしかった。といっても面白いんだよな，このゲーム。特に対戦は燃える。こうなったら，1MBユーザーの人はメモリを増設して遊ぶしかない。よし。

（評価：満点5）
完成度：◎◎◎◎◎
キャラクター：◎◎◎◎
おしゃべり：◎◎◎
ガッツの勝利（2プレイ）：◎◎◎◎◎

THE SOFTOUCH

●ルーンワース〜黒衣の貴公子

マルチエンディングがうれしいRPG

Kaneko Syunichi
金子 俊一

T&Eの最新作は，マルチシナリオタイプのRPG。今度の勇者は山賊に育てられた口の悪い好戦的な少年という設定。う〜ん，やはりRPGとはこんな少年も使いっぱにしてしまうものなのね……。

X68000用　5"2HD版2枚組　8,800円(税別)
T&E SOFT　☎052(773)7770

私が被害車の運転手，金子と申します。最近，ぶつけられ癖がついてしまって困ってます。おはらいでもしようかな。

さてこのゲーム，一般的なARPGと比べて特筆すべき点は，経験値というものの存在がないということ，マルチシナリオであるといった点でしょうか。経験値がない点は，レベルアップアイテムの導入という方法で実現しています。とはいえ，宝箱に入っているアイテムでレベルを上げてしまうのは，かなり強引なような気もしますが(たいていの人は宝箱を全部開けるしね)。

そして，マルチシナリオということでRPGにありがちな“お願いします攻め”を表面的に回避することもできます。たとえば，“○○を取ってきてください”といわれたとき，NOを選択すると相手は引き下がってしまいます。が，その場合次に何をすべきか途方に暮れてしまい，結局は引き受けたほうがよかったりする場合もあります。

さらに，レベルを上げるためにできる限り多くの冒険をすることになると思いますので，結局このゲームはごく普通のRPGとして成り立っているようです。

山賊の子はヨーグルト?

ウェーデル山脈の大いなる峰々が，神々しい朝焼けに染まる。朱光の毛織は瞬く間に山腹を駆けおり，大地に生命の息吹をもたらしてゆく。広大な針葉樹の森は，朱から深い緑へと彩りを変え，その偉容を現し始める。ここは，神聖サイア王国とウェイデニッツ公国の国境近くの森林地帯。山脈越えの旅人が，その疲れた体と心を癒す静かな土地である。新緑に萌える大地を……。

「うるせぇ！」

俺の名前は雪印ナチュレ。こんなふざけた名前を付けやがった親父は，まがりなりにも山賊の頭領だ。ディトゥールの牙っていやぁ，ここらじゃ知らないヤツはいねぇ。気の短さなら天下一品，親父譲りともいわれているが，親父にも負けねえつもりだ。

ケイズニク：若，こんなところにいやしたか。さっきから，お頭が呼んでやしたぜ。

ナチュレ：なんだ？　また昨日の喧嘩の続きでもやろうってんじゃねーだろうな。あの，こうるさいおっさんも，いい加減元気なやつだな。

ケイズニク：まあ，どっちもどっち，お互いさまじゃないんすか？

ナチュレ：なんだとーっ！

ちっ，まっいいか。ケイズニクの野郎は，あとでゆっくりと締めてやるとして，とりあえず親父のところにでも行くか……。

ガラン：いよう，来たな，ろくでなし。今日はな，おめえが生まれてからちょうど17年目。それを祝ってやろうてんだから，ちったぁ感謝しろ。ほら酒だ，酒だぁ。

ナチュレ：あんだぁ？　祝宴だぁ？　てめー，昨日の仕返しにこの俺をハメよーとしてんだろ。どっこい，こんな子供ダマシにひっかかるよーな俺じゃねぇぜ。

ガラン：何をいってやがる，おめえも17，これからは，いっぱしの男として働いてもらわにゃならん。親父としてこんな嬉しいことがあると思うか？　え？

ナチュレ：…いいたいことはそれだけか？

ガラン：そこで初仕事だ。なぁに，たいしたことじゃねえ，セルトレの町にあるしけた教会から剣を1本盗ってくりゃいいんだ。

ナチュレ：ああ？　剣だぁ？　てめー，俺をナメてんな。おうよ。まあ見てろってんだ。ほんじゃあな。

ガラン：こら，まてぃ！　コリノ爺さんのところで装備を整えてから行くんだぞ！

まったく，こうるさいじじいだぜ。ま，そりゃおいといて。よし，装備を整えたらケイズニクを締めにでも行くか。

ナチュレ：おう，ケイズニク，さっきはよくも馬鹿にしてくれたな，成敗してやる。

ケイズニク：勘弁してくださいよ。そうだ，かわりに盗みのコツってやつを教えてあげるってことでどうです？

なるほどねぇ。盗みの道も奥が深いのねぇ。おかげで秘風剣サータルスを，楽に教会からいただけたぜ。レベルも上がったし，まあ首尾は上々ってやつだな。あとは，あのクソ親父に俺の実力を見せつけてやるか。さぁて，ザノバ砦にでも帰るとするか。

山賊の子は運命の子

おや？　女の子が僧衣の男たちに囲まれている。新興宗教の勧誘，いや新手のナンパかもしれないな。今年のトレンドは宗教だって，誰かがいってたような……。

少女：きゃ〜〜〜！

ちょっと違ったみたいだな。ふっ，俺の

そんなに力いっぱい名のらなくても……

出番だぜ！

少女：助けてください。

ナチュレ：あとでメシに付き合えよ。

少女：あなた，誰？

ナチュレ：俺？　俺は雪印ナチュレだ。

仮面の男：ぶわははは，雪印ナチュレ！　面白い名前だ。お前の名前に免じて，今日のところは引き揚げてやる。

ナチュレ：お前の名前は？

仮面の男：暗殺魔術団＜ヴィーフォ＞の神団長，レシェル・カロスだ。

ナチュレ：そうか，お前がジョーのパンチで廃人になっちまった……。

少女：それはカーロス・リベラ。

ナチュレ：そういう君の名は？

少女：ミリム・ルセアンよ。

ナチュレ：電話番号は？

ミリム：この世界に電話はないのよ。

ナチュレ：いや，いつもの癖で……。

が，砦に帰った俺を待ち受けていたのは，レベルアップでも祝杯でもなく，血だらけの仲間が大地に横たわっている，まさに地獄絵図のような光景であった。

ナチュレ：なんだよ，これは！　親父ィ，シャレにしちゃきつすぎるぜ。

ガラン：黒い布の男たちが現れて……。

ナチュレ：わかった，もう話すな。

ガラン：ばかやろう，本題はこれからだ。お前はわしの本当のせがれじゃねえ。

ナチュレ：げろげろ，橋の下で拾ってきたってのは，ほ，本当だったのか？

ガラン：ばか，お前が手にしているその剣の精霊が，17年前お前を連れてきたのだ。

精霊：私は秘風剣サータルスの精霊。この子は運命の子，世界を光に導く運命の子だ。

ガラン：じゃじゃじゃじゃ～～ん。

精霊：…それはベートーベンだ。ともかく，この子を頼んだぞ……。

ガラン：そうして，おまえをわしに託した精霊は消えてしまったのだ。

ナチュレ：どこまでが本当なんだ？

ガラン：すべてだよ。わしはもう寝るからな，がんばって世界を光に導けよ……。

ナチュレ：親父ぃ～～！

山賊の子は旅に出た

親父や仲間の仇をとるべく，俺は旅に出た。なぜかミリムも一緒だ。暗殺魔術団＜ヴィーフォ＞の噂を求め，サルトレの町に着いたはいいが，相変わらずシケた町だぜ。まあ，噂話ってのは酒場に集まるって相場は決まってるからな。夜になるのを待って俺たちはドリオル酒場に行ってみた。

アル：おお，ミリム様ではありませんか。

健康状態が“泥酔”ってのもすごいな

ミリム：あらアルじゃない。

アル：その横にいる小汚い山賊みたいな野郎は何者ですか？

ナチュレ：山賊とはいってくれるじゃねえの（当たってるけどよ）。

ミリム：まあいいじゃないの，旅は道づれ世は情け，火事と喧嘩は江戸の花（注1）っていうでしょう。アルも私たちと一緒に旅に出ましょう。

仲直りもかねて他の店で飲みなおすことにした。ところがどっこい，この町には酒場がひとつしかない。どうしてもはしご酒がしたかった俺たちは，サイアの街にやってきた。町ではなくて街というところにこの街の大きさを想像してもらいたいね。ただひとりミリムが反対したけど，この辺じゃいちばんの街だし酒場ぐらいあるだろう。

ナチュレ：この店で飲みなおしだぁ。

ミリム：わかったわよ。こうなりゃとことん付き合うわよ。ングングング，プハ～。

ナチュレ：お前酒乱だったの？

ミリム：酒乱とはなによ，私と勝負しようってのね。え～いぽかすかぽかすか……。

ナチュレ：このやろ～～ぽかすか……。

警察：…トラ箱行きなさい，あんたたち。

治安警察に捕まった俺たちはトラ箱に入れられちまったんだ。

ミリム：なんでこんなとこにいるのよぉ。

ナチュレ：おまえ，なんも覚えてないの？

ミリム：なんのこと？

ナチュレ：ほんっとぉに，なんも？

ミリム：だから，なにをよぅ。

ナチュレ：酒場で暴れ回ったの！　覚えてない？

ミリム：この上品なあたしがそんなことするわけないじゃない。

ナチュレ：…こいつ本格的なトラだぜ。

アル：姫様，迎えに上がりました。

ナチュレ：姫様ぁ？

カナン王：おお，そなたじゃな，娘を救ってくださったのは。礼を申すぞ。

ナチュレ：はぁ。

カナン王：そこでだな，ついでといってはなんではあるが，エリスの首飾りを捜してきてはもらえぬだろうか？

ビジュアルシーンにもいろいろパターンがある

ナチュレ：やっとRPGらしくなってきたか。

カナン王：何かいったかの？

ナチュレ：いや，こっちのことだ。まあ旅のついでに捜してやるよ。どど～んと大船に乗ったつもりで待っててくれ。

カナン王：それはありがたい。それではつまらない物ではあるが，旅の支度をさせてあるから受け取ってもらいたい。

なんだかんだいってレベルを上げた俺はエリスの首飾りを捜す旅に出た。ああ，これから使いっぱ（注2）の真髄を極めるわけだな，こりゃ。やっぱ，RPGの主人公は使いっぱだよなぁ。

注1：昔の偉いお奉行様（記憶では大岡越前）が本当にいった言葉。正しくは火事と喧嘩は江戸の恥である。いつからか江戸の花といわれるようになった。

注2：ぱしりともいう。クラスに最低1人はいる。使いっぱの君，落ち込まないで自分をRPGの主人公だと思おう。少しは気休めになるかもよ。

総評

このゲームのタイトルであるルーンワースはこの世界そのものの名前なのです。

で，パッケージを開けると永久保存版データハンドブックなんてものも付いてくるんだけど，これが使えない。その中にむちゃくちゃ細かい設定が書いてあって，その上にこの話が成り立っているのだけど（ようするに黒衣の貴公子編ってわけね），実際に活用される部分は，最後に載っているアイテムガイドだけ。初めて読んだときは，あまりのめんどくささに途中で寝ちゃったもんね。

このおまけの内容にちょっと凝りすぎた分だけ，実際のゲームの奥が深く思えてこなくなっているのではないかな。結局はマニュアルプロテクトで終わってしまっているのは，かなりもったいない。バックストーリーはプレイヤーに想像させるほうが楽しいと思いますが，いかがなもんでしょう，T&Eさん？

	（10段階評価）
イース度	7
音楽	5
シナリオ	7
グラフィック	6
オーガスタへの期待	10

THE SOFTOUCH

●闇の血族

美少女探偵は超能力者?

Komura Satoshi
古村　聡

読ませるアドベンチャーゲームとでもいうべきノヴェルウェア。やっとこさ登場の新作はミステリアスアニメーションアドベンチャーという思わず舌を噛んでしまいそうなジャンルの「闇の血族」です。

X68000用　5"2HD版4枚組　8,800円(税別)
システムサコム　☎03(635)7609

当初の予定よりちょっと遅れて登場のこのゲーム（1カ月やそこらなら遅刻のうちには入らんというのがこの業界の掟だったりもするんだけど)。サコムのノヴェルウェアの5作目にあたるわけでありますが，今回はいままでとはちょっと趣が変わって映画仕立てのゲームになっているのであります。まさか，スタッフが締め切り間際まで映画ばっかり見てたせいで納期に間に合わなかったんじゃ……，などと思ってしまう疑い深い私なのであります。

ま，それはそれとして実はこれで結構映画好きな私のことでありますから，喜び勇んでこのゲームを待ち望んでいたのであります。

この「闇の血族」なのでありますが，いままでのノヴェルウェアでの，やれ近未来にドームができた，38万キロのエレベータができた，というようなSFチックな話とは180度,いや，まったく違って，なんと現代のとぉきぉのアパレル娘（決して暴れる娘ではなひ）が主人公。**美少女名探偵♡魅由**の繰り広げるミステリアスアニメーションアドベンチャー第1弾！ なのであります。艶やかなファッション界を襲う奇怪な殺人事件。南米の血に隠された秘密とは？そして魅由を待ち受ける血族の宿命は？

……なお，これらはパッケージに書いてあるままであって，これらの問題が解決しない，あるいはどこでそれらのことが解決したのかよくわからなくても，私のせいではありません。

さあて"闇の血族"試写会だ!

というわけで，こちらは「闇の血族」試写会会場前です。寝袋に入った人たちがごろごろと並んでいる光景はまさに壮観なのであります。一応，念のために言っておきますが，本当に試写会があったわけではありません。映画風に仕立て上げられたゲームなので，試写会としゃれてみただけであります。

あっ，かあいい受付のおねーさん，このごろごろとまるで築地のマグロのように転がっている人たちは，やっぱりこの試写会を見るために並んでいるんでしょうか？

「いいえ，この人たちは制作スタッフです。昨日，やっと今日試写するものが出来上がったんですけど，すぐにパート2のほうの制作に入るので試写会が終わるまで，ここで仮眠しているんです」

うーん，たしかに試写会の招待券には#1（第1話という意味）と書いてある。そうか，すぐにパート2が出るのか……。そこはかとなく一抹の不安が脳裏をよぎるのであります。

さて，右手にポップコーン，左手にコーラのコップを持って席に着くと，いよいよ試写会の「はじまり，はじまり」というわけであります。

すうっと，毎度おなじみのサコムのマークが消えると今度はNOVELWAREの文字がずずずずっつと出てきます。うーん，最近は拡大・縮小が業界中の流行なのであろうか。ずずっ。

どっかのテレビドラマ!?

オープニング。

東京・新宿の夜景。靖国通りだろうか。伊勢丹が見える。

画面は横にパンしていく。

黒のマネキン人形が帽子を被った横顔をショウウィンドウのガラス越しに見せる。

首のスカーフにはシャネルのマーク。

夜の道路に赤いテールランプ。

遠く，西口のほうには高層ビルが見える。三井，住友，そして左の少し小さなビルはNSだろうか……。

そして，タイトル。"闇の血族"の文字……。背景はダウンスクロール。

オープニング終了。

ここで，魅由の自己紹介。

サンプリングの声が流れる。

彼女は新宿にあるデザインスタジオYoの新人アパレルデザイナー。

マンションの朝。ガラステーブルがある。ソファーやステレオも。カーテンが開いて光が差す。遠くから小鳥の声。そして子供の声。

ここは魅由の部屋。出窓のピエロ。白い机に椅子。白いクローゼット。

父親がこのマンションのオーナー。彼女は世間でいう「お嬢様」ってことになるわけ。

……だそうである。いえ，別にいいんですけどね。なんか，女の子の独り暮らし，そして東京のド真ん中でワンルームではな

危険なイメージが赤い色で……

いマンション（ワンルームだって月10万円じゃきかないし）という優雅な生活。思わず，浅野ゆう子とかが出てきそうな気がしてくる。カチ，カチ，カチ（マウスをクリックする音）。

んで本筋はというと

魅由の持つ力とは……。桁外れの勘の鋭さ，常人にはない強い第六感。調子しだいでは未来予知もできる。電話のベルが鳴った瞬間に誰がかけてきたかもわかる……。

電話のベルが鳴った。その瞬間，危険なイメージが脳裏を走った。

マリーが死んだ。唯は泣きじゃくりながら電話口でそう言った。マリーが殺されたと。

マリー・富岡。通称クイーン。華麗な男役の似合う，メキシコと日本のハーフでスタジオYoの専属モデル。魅由の4つ年上。マリーの死体の中には赤い血がなく別のものが詰まっていた。なぜマリーが？

スタイリストのリーンとともに魅由はスタジオYoにやってくる。なにか異様だ。中原さんもいる。……なに？　なんで泣いているの？　みんな，魅由たちが入ってきたことに気づかない。

「唯が……，唯が殺されちまったよォッ！」

……魅由はその夜，夢を見た。遺跡のようだった。夜の森。そして湖。……建造物……見たことがある。あれはマヤ文明の……ユカタン半島の……神殿，そして彫像。こっちを向いている。なにか神がかった彫像。

そして，どこかから声が，

「……流浪の民百千が集い，その姿宇を神へと変容せしめしこの地に汝の真の姿ありしや……」

最後のなにか，詩の一節のような声。その声は魅由自身のものだった。これはいったい……（「姿宇」ってなんだろう？）。

ここがスタジオYo

一瞬，女かと思った

そして朝。魅由は喫茶FELICSへと向かった。マスターによると，唯の直接の死因は刺傷からの大量出血。頭部，心臓，下腹部をメッタ突きにされていた。そして，

「唯の死体の血も赤くなかったそうですよ……」

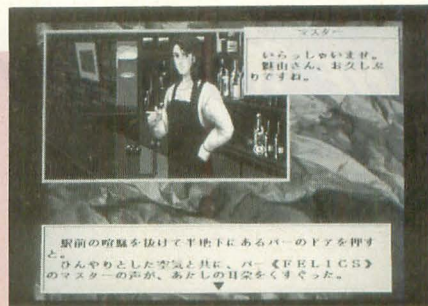
唯も身体中の血液がなく，別のものになっていた。死体の周りには赤い血が飛び散っていたというのに。

さーて，来週の○○○○○は!?

まあ，さわりだけ紹介すると，こんなお話なわけです。

しかし，同じノヴェルウェアの「ドーム」，「38万キロの虚空」でも思ったんですが，ちょっと設定が消化不良じゃないでしょうか。あれだけ，いろいろ設定を細かく作っていながらそれが十分に生かされてないのはとても残念です。

つまり，どういうことかというと，「魅由は新宿のスタジオYo」のデザイナーなんでしょ。なぜ，作者は新宿ということにしたのか？　あれだったら南青山でも表参道でも関係ないんじゃないでしょうか？　それになにか新宿と南米を結ぶ線があってもよかったのではないかな。

たとえば，NS地下や小田急で南米マヤ文明展がある。そこで，そういえばおばあちゃんがブラジルだっけと思い出させる（あるいは南米方面には誰と誰がいるというのを思い出させる）。昼間の東口の雑踏のなかで例の詩の文句が聞こえる。え，なに？　空耳？　という効果が使える。真夜中の誰もいない新宿地下道で例の詩が聞こえてくる（今度は結構不気味なイメージが出る）。そして，成田空港で「そうだ，メキシコへ行くんだ！」と思わせる。

こういういう感じで設定をうまく使えばいろいろなやり方で演出できるのに，ただ夢の中で1回だけ出しちゃう。ちょっともったいないんですよねえ，やっぱり。そんな細々した設定は作らなくてもいいんだから（作ってはいけないとはいいませんが），1つひとつの設定を生かすことを考えてストーリーを作るようにしてほしいと思います。

さて，今回の話は前編ということで，なんと次回の予告もついています。これがまあとんでもなくよくできているんです。もうサンプリングあり，拡大・縮小ありのありありモードでほとんど映画のよう。あまりによくできているんで，もしかしたらこの予告編を見せたいがためにこのゲームを作ったんじゃないかと思ったりしました。はい。

それと，よくできているといえば，オープニング。私が新宿を知っているからそう思うのかもしれませんが（あるいは画像取り込みだから当たり前なのか）新宿という街の無機質な感じがよく出ていて，設定が生きている部分だと思います。

さて，ストーリー的にも，まだ，序盤という感じで終わってしまったこのゲーム，どういう人物かわからないうちにみんな殺されてしまったのはご愛敬としても，この先，どう展開するんでしょうね。そうやって考えると，このペースだととてもあと1回くらいじゃ終わらないような気がするんですが，完結編としっかり予告に書いてあるし，とりあえず次回に期待ということですね。

それではみなさん次回までサヨナラ，サヨナラ，サヨナラ（やると思ったでしょ）。

厳しいかもしれないけど……

そういうわけで，このゲームはこのあと発売予定の完結編あってのパートⅠで，現時点での評価はちょっと避けたいと思います。内容に関する評価は完結編が出てからということにさせてもらいましょう。

それにしても，ディスクに入りきらない，あるいはシナリオが大きすぎてひとつのゲームにはできなかった，というのであれば前後編の2つに分けるのではなく，無駄なところや本筋とはあまり関係ないところを削って1話にまとめるべきだったと思う。前後編に分けた結果，無駄な描写や表現（前半部の自己紹介，シャワーのアニメーション）が多くなり枝葉の部分ばかり見せてしまい肝心の幹の部分がぼやけてしまっているようだ。

自分たちの作った，設定・脚本・プログラムがもったいない，かわいい，出してやりたいという気持ちは私もとてもよくわかる。が，ゲームをする人が気持ちよくゲームを終了するために，あえて自分の書いたシナリオを削りデータを打ち捨てる。たとえ，結果的に自分の作ったものの半分，1/3しか見せることができないことになってしまっても，それこそがプロのなすべきことであり，プロの厳しさなのではないかと思うのだが。

完結編に期待しよう。

THE SOFTOUCH

●提督の決断

提督の艦隊 沈黙の決断

Ogikubo Kei
荻窪 圭

光栄の鬼才，シブサワ・コウの新作は太平洋戦争を舞台にした本格派シミュレーションゲーム。君の冷静な判断力で実際の歴史では負けた戦いをも勝利へと導こう。まあ，連合国側でやるのもいいけどね。

X68000用 5"2HD版3枚組 14,800円(税別)
光栄 ☎045(561)6861

なにやら8月15日は終戦記念日だそうである。私は生まれてなかったから知らないが，毎年「連合艦隊」やらなんやらの和製戦争映画をテレビが熱心に見せてくれるので戦争があったことくらいはわかる。でも，「あーそーゆーもんだったんだね」とか感じるだけで，もはや他人事である。

しかし，その他人事感が日本の豊かで暖かな戦後教育によるものだと知ったのは，台湾の女の子たちと飲んだときであった。あのときは日本語を喋れる留学生と，日本に遊びにきた英語と中国語しか喋れない元新聞記者と，秋葉原にハンディカムを買いにきた中国語しか喋れない旅行者と，日本語しか喋れない我々が入り混じって，“英語漢字混じり文”という恐ろしい筆談までしてしまうシュールな体験をさせてもらった。そのとき戦争の話題を持ち出したのはこっちであった。気になっていたからだ。そして，台湾の人々（すべてかどうかはわからないが）にとっては他人事ではないということだけはわかった。

他人事かそうでないかはさておいて，侵略戦争はあったのである。核爆弾を落された日本が原爆の日を記念するように，被害者は常に忘れないようにしている。終戦記念日くらい“もう侵略はしないぞ集会”でもすればいいのに，と思うのであるが，誰もやんないだろうな。鶏の頭なみに素早く忘れたがっているらしいから。

そんなことを思いながら，「提督の決断」を立ち上げた私はCRT上に美化された太平洋を侵略していくのであった。

最初が面倒，最初が肝心

「提督の決断」は例のシブサワ・コウであるから，おなじみ，キャラクタのパラメータ設定から始まる。山本五十六などといった有名人が居並ぶ中，スペースキーを押す。勝とうと思ったら，特化才能人材をたくさん作る必要がある。作戦会議用に作戦能力の特化したやつ，艦隊戦用に艦船能力の特化したやつなどなど。全部で120の数値を4つに振り分けるのだが，そのうち勇敢さ（こんなもんは役に立たない）以外のどれかを80以上にしたいところだ。んでもって，作戦会議担当はひとりでいいので，あとは艦隊を任せられる将校にすべきだ。シブサワ・コウゲーム全般にいえることだが，「最初が肝心」なので時間をかけねば。

シナリオはあの有名な“リメンバー真珠湾”から“ミッドウェイ”，“最期の奉公，大和沈没”までいろいろある。が，私としては最初の“日米決裂”シナリオをお勧めする。これは“太平洋戦争すべてを最初から最後まで自分でやる”という地獄のシナリオだ。そのかわり，史実を睨みながら負けるとわかっている作戦はしなくて済むし，じわじわと戦況をコントロールできる。しかし，時間だけは飽きるほどかかる。夏が終わってアキが来るくらい時間がかかるぞ。心しておくように。

で，母港の呉から話は始まる。プレイヤーは海軍司令官で，陸軍はコンピュータが勝手にやる（ろくなことはしてくれないが）。最初はたくさんの艦船とたくさんのお金とたくさんの暇があるわけで，ここで楽をしては楽しめない。次に述べるようにすれば，明るい艦隊戦に臨めるというもんだ。

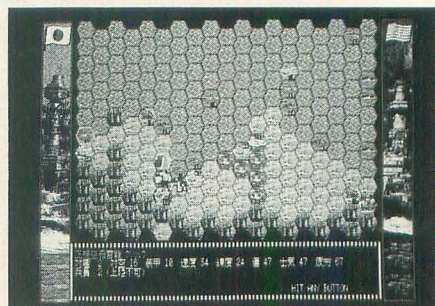
まず，艦隊の編成。デフォルトだと第1艦隊に空母が集まっているから，第1艦隊は空母4隻くらいにして，残りは別の艦隊へ移す。んでもって，各艦隊は敵基地攻略用と味方基地防衛用・支援用に分けて編成し，決して中途半端なものは作らない。

編成が済んだら，補給をする。一度に全部やると20日以上かかるので，最初に出撃する3艦隊くらいにしておく。

作戦会議を開いて重点作戦目標を決める。作戦能力の高いやつを会議に出すと希望が通りやすい。作戦能力が低いと陸軍の提案を飲むはめになり，それは敗北への道だ。間違ってもハワイなどといってはいけない。まずは，原油のとれるソロンやサンダカンだ。

補給が終わるまでやることがないからといって寝ていては駄目である。会議を開いて技術開発なり兵器生産なり（航空機は絶対に足りなくなるので，たくさん作るべし）工作員の派遣なり外交政策なり，とにかく怠慢しては勝てない。

補給が終わったら，残りの艦隊を補給に回し，補給済み艦隊は作戦開始である。第1艦隊は一度出港すると戦果を上げて戻る義務があるので，重点作戦目標へ突き進む。

最初は敵も元気なので，こちらの基地を狙ってくる。敵艦隊の攻撃を受けたらそこ

港へ攻め込んだところ

へ艦隊を直行させよう。敵艦隊には雷撃機。頑張れ空母！　である。

歴史なんか狂わせろ

そんなこんなで，史実に従った宣戦布告がなされる。でも，真珠湾なんか行かない。

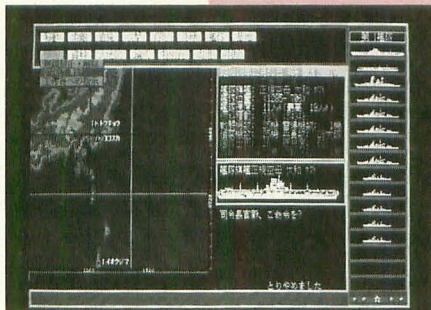
前半は余計な領土拡大を狙わず，じわじわと，うろうろする敵艦隊を叩きつつ，富国強兵殖産興業文明開化滅私奉公……。

海で出会うなり基地へ空撃なりして始まる戦闘は，相も変わらずHEX戦。でも面倒臭くはない。なぜかというと，自分で戦う必要はないから。第１艦隊以外は勝手に戦ってくれるし(それがうっとうしいときはHEX OFFにすると結果だけ教えてくれる)，第１艦隊だって司令官に任せておけばいい。もっとも，コンピュータは往々にして間抜け（というより戦闘のアルゴリズムが目の前の戦果を求めるよう組んでるみたい）なので，「この大バカ者！」と歯を噛みしめるのは精神衛生上よろしくない。こちらが圧倒的戦力のときを除いて，第１艦隊くらいは自分で戦いたいものである。

しばらくすると，史実に従って大和やら武蔵やらがいろいろと完成する。私のように大和を空母に改造するなどとやって遊んでいたら（空母大和！　……私の歴史では“宇宙戦艦ヤマト”はなく“宇宙空母ヤマト”なのだ)，あとで工業力が足りなくなるので注意。

敵さんはある程度戦力をずたずたにされると，ハワイにこもるようだ。だから，そうなったら（昭和18年くらいかな？）連合国の（つまりアメリカの）ハワイに向けて富国強兵。技術力は上がっているはずなので，艦船を修理に出してパワーアップしたり，新兵器が開発されたらそれを搭載できるように改修したりと仕事は多い。

でもって，東南アジアを野放しにしておくと，いつの間にか敵基地からの攻撃を受けて陥落したりするので，基地への補給は忘れずに。空母に雷撃機を積んだ小艦隊を敵輸送船襲撃用に洋上に置いておくと便利である。

私なんか富国強兵政策の一環として，ソビエトやタイと同盟を結んでしまった。イタリアはもう降伏してるし，歴史はシッチャカメッチャカである。なんてったって，昭和19年というのに，日本はジェット戦闘機を開発してる（陸軍がぶつぶついうのも構わず，航空技術に金を注ぎ込んだのだ）し，ミッドウェイは傘下だし，ガダルカナルも占領下，ホンコンも領土という具合なのである。

僕の大事な空母大和

で，ハワイへ総攻撃をかけ，敵艦隊を殲滅すれば“勝利”であるが，それがなかなか大変（当たり前だ)。

ヨーロッパでは史実どおりにノルマンディ上陸はするわパリは陥落するわで，ドイツが降伏する前に頑張っておかないと孤立無援になってしまうので注意だな。

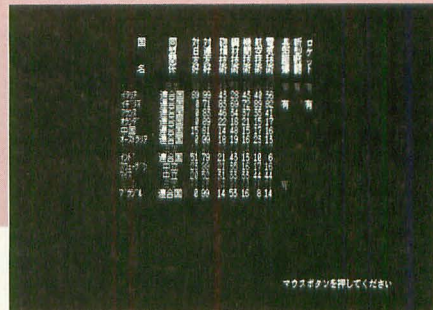
こうしてこうしてこうして，戦況は泥沼となり，原稿は進まず，ほかの仕事にまで影響が出て，エアコンのない部屋でX68000と私の脳味噌は熱暴走するのであった。

歴史SLGは教育か？

というわけで，義務は果たした。ここからは好きなことを書く。

あー，かったるい（いきなりですみません)。

こういうゲームは，戦闘しているときと次の作戦を練っているときが一番楽しい。しかし，一番大切で一番時間のかかるのが作戦と戦闘の間なのである。完全修理には何カ月もかかるし，第１艦隊はいちいちマウスで少しずつ移動せねばならないし，マップはいちいちディスクに読みにいく。

何がいけないのか考えてみると，プレイヤーは戦況を見ながら「次はどこを攻めよう」とか「次の作戦はあそことあそことあそこだ」などと考えるのである。で，「行くぞ！」「おう」となるのであるが，「次はこうするぞ！」と「行くぞ，おう！」の間の準備にやたら手順があるのだ，悠久の時間が流れるのだ。疲労の原因である。

ほら，日本とソ連とタイの三国同盟

光栄のゲームはE&E。「エネルギー&エレクトロニクスの東芝」ではなくて，「エンターテイメント&エデュケイション」だ。つまり，遊びながら勉強するというありがたいお題目だ。その陰で歴史シミュレーションを飽きもせず送り出してきた。

しかし，である。この歴史シミュレーションシリーズが何を生み出したかというと，「歴史，好き？」と問われて「はい」と答えようもんなら「信長と秀吉と家康で誰が好き？」とか「三国志がどうとか，かんとか？」という風潮だ。これは光栄のせいではなくて，うっとうしい世間での風潮なのだが，いつから「歴史」＝「日本の戦国時代」or「中国の戦国時代」になってしまったのだろう。わたしゃどっちにも興味はないが，歴史は好きである。特に古代ローマやヨーロッパの中世史が好きである。勝手に歴史の範囲を狭めてもらいたくないのである。光栄さん，次は「ガリア戦記」でもやりませんか？

でもまあ，単なる戦争シミュレーションが「エンターテイメント&エデュケーション」なのだから，いい時代になったものだ。

この道はどこへ行く道

光栄の歴史シミュレーションシリーズも「信長の野望」スタイルを継承したまま，とうとう「明治維新」どころか，「太平洋戦争」まで来てしまった。この道はどこへ行く道，だ。

「提督の決断」もX68000用にBGMや効果音がついたり，フルマウスオペレーションになったりしたが，良くも悪くもシブサワ・コウそのものである。ついでにRAMディスクやハードディスクへのインストール機能もほしかったところだ。ディスクをガシガシ読みにいきすぎるから。

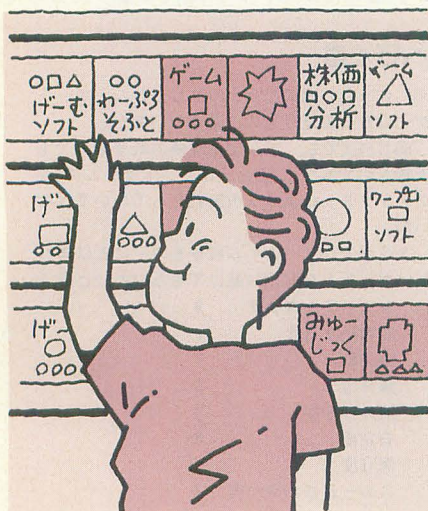
で，そろそろHEXを使った大規模戦争ものにみんな飽きてもいい頃ではないか。たとえばアートディンクは同じ太平洋戦争でもHEXを使わない，指令だけすれば米粒ほどの艦隊がノソノソとCRT上を動いていくお得意の箱庭観賞型シミュレーションを（成功したかどうかはともかく）作った。HEXというのはしょせん，ボードゲームからの借りものに過ぎないのだ。

同じHEXでも，こんな大規模の戦争ものでも大陸ものでもなく，たとえばボードゲームの「タクティクス」のような市街戦だっていいではないか。

まさか次が「ベトナム戦争もの」にとは思わないが，そんな泥沼は避けてもらいたいと思う。

シブサワ・コウ度	9
終戦記念日度	8
HEX度	7
操作性	6
病みつき度	8
安定度	8
保守度	10
スカっとさわやか度	3

AFTER REVIEW

ゲームにもいろいろあって，お手軽なものもあれば，じっくりやらなきゃいけないものもあるんですよね。あなたはどちらがやりたいですか。まあ，両方を交互にやるというのが一番いいかも。

ギャラガ'88

▶今月，(記事の中で)一番わらけたので。　愛知県・志賀　宗一(17)

▶電波のオリジナル(アレンジモード)が気に入ってしまった。栃木県・毛塚　健次(17)

▶むしょうにマッピーか，ラリーXをやりたくなる。　栃木県・鹿又　健(21)

▶音楽がよかったから。　東京都・辻　康介(16)

▶なにも考えずに楽しめる。グラフィック，サウンドもよいほうだ。　京都府・岡田　伸一(22)

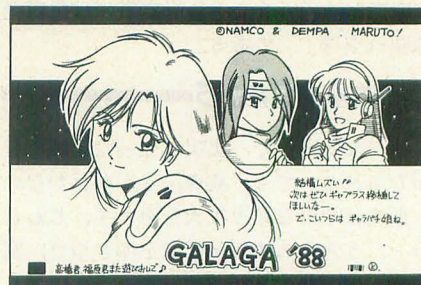
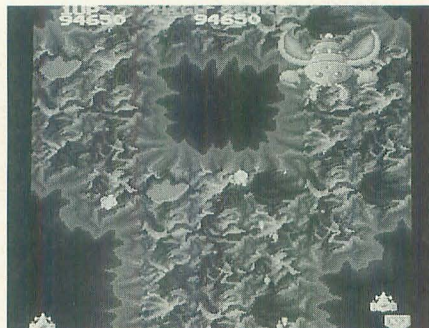
▶な……なつかしい。東京都・竹川　忠(17)

▶X68000のはやったことないけど，昔からギャラガが好きだから。　福島県・捧　宏太郎(18)

▶いまさら，なんでか面白い。　神奈川県・中村　雅彦(22)

▶ギャラクティック・ダンシングがいい。　静岡県・山下　禎久(17)

▶X68000で一番手軽に楽しめるソフトだと思う。　兵庫県・木下　達也(18)

▶撃って，撃って，撃ちまくる快感！　やめられまへ～ん！　三重県・竹腰　敦司(19)

▲神奈川県・丸藤　俊之

▶画面がきれいだから。デンパだから信頼があるから。　岐阜県・高橋　賢(18)

▶FM TOWNSでも，PC-9801でもないというゲームあるじゃない。これがそう。　大阪府・遠藤　勇(33)

▶このゲーム，うちの父さんのお気に入りとなってしまったようです。　静岡県・小澤　健一(16)

このギャラガ'88はPCエンジン版でやったという方も多いことでしょう。あれもなかなかよくできていたけれど，X68000版はさらによい出来。なんか高級感が漂っているんですよね。本体が高いからというわけではないでしょうけど。やっぱり，音とか絵の鮮明さが違うからでしょうか。効果音もほとんどサンプリング音だし，ちょっとしたしゃべりも入っている。ボスのアイキャッチシーンもしっかり入っているしね。

しかし，3機合体したあとでやられてしまうと，もうやる気がなくなってしまって思わず電源を落としてしまう……。あきらめが早すぎるんでしょうか。それにしても2機合体は役に立たない，まあそこがいいっていう話もありますけど。

X68000用　5"2HD版2枚組　8,200円(税別)
電波新聞社　☎03(445)6111

発売中のソフト

★三国志II

X1用で先行発売されていた三国志IIが，お待ちかねX68000にも登場だ。劉備，曹操といった英雄になり代わり，数々の外交手段や計略，戦略を巡らして中国全土の統一を目指す。X68000版では漢字も読みやすくなり，メイン画面にコマンドメニューが出たり戦闘時にグラフィックが表示されたりと一段とクオリティアップしているぞ。

X68000用　5"2HD版3枚組　14,800円(税別)
光栄　☎045(561)6861

★セレクテッドソーサリアン5

TOP10でも息のながーい人気を保っているソーサリアン。読者からのシナリオコンテスト優秀作シリーズも，今回で一応完結。賞金1万$を賭けて行われるレース「それゆけ！　ドトーのトライアスロン」と，脱獄したワーウルフを捕らえる「封印」の2作のシナリオを収録。

また，マガジンディスクにはNEWドラゴンモードもついているのでお買い得かも。

X1turbo用　5"2D版2枚組　2,900円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

★ヴァーチュア・ナイト

遥か昔，すべての生き物を滅ぼそうとした魔人がいたが，とある魔術師が聖なる武器によって異世界へ封印することに成功した。いま再び世界の異変を感じ取った魔術師の後継者マスダミ・アーはその原因を探るため旅に出る。ログインのソフコンで3rd Prizeを獲得した横スクロール型アクションRPG。6種の魔法の画面効果がウリ。

X1用　5"2D版2枚組　2,000円(税込)
ブラザー工業(TAKERU)　☎052(824)2493

★FLY

ログインソフコン入選作。慣性力が働く円形の自機を操作し，コースを3周してタイムを競うアクションゲーム。ルールはシンプルだが，ツルツルのコースから落ちないように回ろうとするとなかなか難しい。好きな面をプレイできるモード，1面から順番にプレイしていくモード，ビデオ機能などがついている。

トンネルズ&トロールズ

▶初めはかったるいが，やり込むと面白くなってくる。　静岡県・笠井　博一(20)
▶テーブルトークRPG派なので１票。　東京都・國竹　泰天(16)
▶D&Dがキライだから。　京都府・岡本　直樹(17)
▶根気を養わせてくれるから。　愛媛県・本田　和正(16)
▶フロッピー交換の苦労を味わいなさい！　徳島県・菅野　宏和(18)
▶いま，つまってるから（なんじゃそれ)。　兵庫県・中馬　高嶺(19)
▶テーブルトークでもやっていたし，なんといっても装備の変化に伴い画面上の姿も変わるのがいい！　北海道・池内　亮平(18)

じっくり腰を落ち着けてやると，かなり面白そうだけど，時間がかかりそう。そういう本格派のゲームだから一度はまるともうやめられないでしょうね。比較的時間が余っているという人にはぴったりのゲームではないでしょうか。

X68000用　5″2HD版3枚組　9,800円(税別)
X1/turbo用　5″2D版12枚組　9,800円(税別)
スタークラフト　☎03(988)2988

クォース

▶はっ，はまる。いまのところ14×12が最高。　愛知県・魚住　雄一(21)
▶タイトル画面の帽子をかぶった女の子が好きだから。　千葉県・竹沢　裕利(18)
▶対戦モードが面白い。　東京都・木村　亮(17)
▶デモの巨大な四角を私は一生忘れない。　静岡県・清水　健太郎(17)
▶初めて２つめのジョイスティック端子にパッドをつないだ。　宮城県・小松　仁(17)

パズルゲームはもうなんとなく食傷気味なんですけど。やっぱりこういうお手軽なやつにはついつい手を出してしまうんですよね。始めてしまうと結構燃えてしまうし，なんといっても２人プレイがある。まあ，これもブロックを消す快感ということで，ありふれてはいるんですけど。それにしても最初のデモは力が入ってますよね。どんなたいそうなゲームが始まるんだろうかと思っていると，あの超シンプルなゲームが始まる，すごいギャップを感じますよね。なかなか笑える。

X68000用　5″2HD版　6,800円(税別)
コナミ　☎03(262)9110

D-Again

▶昔から（Ⅰから）ブロンウィンが好きっ！　北海道・西崎　貴博(15)
▶ブロンウィン，かわいい♡　京都府・大谷　将友(17)
▶パッケージのブロンウィンがかわいい!!　奈良県・森　芳生(14)
▶パッケージのブロンウィンがめちゃかわいい。　東京都・清水　健年(19)
▶ブロンウィンが好き！　愛媛県・住友　智代(17)

……別にいいんですけどね。なんかもう，ブロンウィンが好きばっかりなんですけど……。内容のほうには意見がないんでしょうか。ストーリーとか，システムなんかも結構よくできているはずなんですけど。まあ，「なにはともあれ，ブロウィンがすべて」ということなんでしょうか。こういうハガキばかりでもしょうがないような気もしますね。ブロンウィンのかわいさのあまり，みんなほかのところには目が届かなかったということにしておきましょう。しかし……。

X68000用　5″2HD版4枚組　8,800円(税別)
データウエスト　☎06(968)1236

X68000用　5″2HD版　2,000円(税込)
ブラザー工業（TAKERU)　☎052(824)2493

★GUNSHIP

アメリカのヘリコプター・シミュレーション「GUNSHIP」がX68000に登場。アメリカ軍の最新攻撃ヘリAH-64Aアパッチを操縦して訓練飛行からゲリラ戦，米ソ間の仮想戦まで，幅広いミッションをこなしていく。任務の達成具合によって得点が与えられる。働きによっては昇進や勲章授与などもあるとか。キー操作も凝っているため，操作がひとめでわかるテンプレートまでついてくるという力の入れようだ。

X68000用　5″2HD版　11,800円(税別)
マイクロプローズジャパン　☎0423(33)7781

新作情報

★MISTY VOL.5

推理の面白さを中心に据えたアドベンチャーゲーム。依頼者の話を聞いてさまざまな場所を訪れ，手掛かりを得て犯人を推理する。不必要なグラフィックやミュージックを排除し，ロープライスを実現している。ユーザーを対象にシナリオ投稿も受付中とのこと。９月21日発売予定。

X68000用　5″2HD版，XIturbo用　5″2D版
定価5,000円(税別)　タケル価格4,000円(税込)
データウエスト　☎06(968)1236

★機甲師団

史上最大の作戦といわれる連合軍のノルマンディ上陸作戦のシミュレーションゲーム。作戦画面で各マップに投入する戦力バランスを決定し，戦場画面で各部隊に指示を与えてドイツ軍を撃退する。隊の移動や戦闘はリアルタイムに進行し，移動コースもヘクス上ではなくラインを描いて指示するなど，アートディンクらしい独創的なアイデアを凝らした作品だ。９月28日発売予定。

X68000用　5″2HD版3枚組　9,500円(税別)
アートディンク　☎0474(77)7541

★プール・オブ・レイディアンス

AD&Dシリーズ第２弾が登場。前回はアクションRPGだったが，今度は本格的RPG。ストーリーの舞台となるのはワランという都市。幾多の盛衰を繰り返したこの街は，人間の住む新市街とモンスターのはびこる旧市街に分かれてしまっている。プレイヤーはこの街で次々とクエストをこなし，やがて旧フランを支配する勢力を打倒することが目的となる。マルチクラス制，６種類のアラインメントなど，システム関係も充実している。10月21日発売予定。

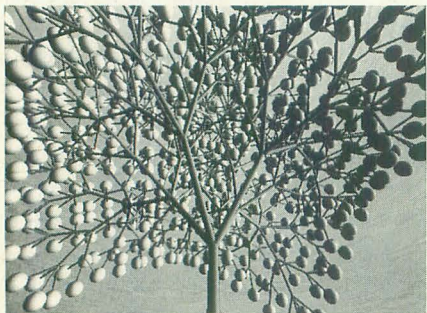
XIturbo用　5″2D版5枚組　9,800円(税別)
ポニーキャニオン　☎03(221)3161

★熱血高校ドッジボール部サッカー編

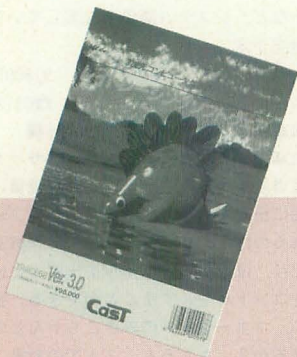
熱血高校のくにおくんが，今度はサッカーゲームで登場。食中毒で倒れたサッカー部員に代わって大会に出場することになったのだ。プレイヤーが操れるのは１人だけだが，事前に作戦を決めておいたり，ほかの選手に指示を出すことによって試合をコントロールすることができる。強豪12校を倒し，マネージャーのみさこちゃんとの約束を果たせるか？　もちろん，２人での対戦，あるいは協力モードもあるぞ。

X68000用　5″2HD版　価格未定
シャープ　☎03(260)1161

THE SOFTOUCH

●C-TRACE68 ver.3.0
●C-TRACE68 TP

高速&超高速レイトレ新次元

Tan Akihiko
丹 明彦

ボクセル分割を加えて高速化C-TRACEのバージョン版とさらにトランスピュータで起高速化したTP版を紹介する。トランスピュータボードの大幅な値下げでCGアニメも夢ではなくなった？

X68000用　5"2HD版98,000円(税別)
(TP版298,000円)
キャスト　☎03(705)1065

C-TRACEのバージョンアップも久しぶりである。バージョン2が発表されたのが昨年の春だから，かれこれ1年半になる。以前トランスピュータ版のレビューをしたのだが，それでも1年前。そのあいだにC-TRACEはユーザー層を着実に広げてきたようだ。

そして1990年秋。満を持しての登場というべきか，C-TRACEはバージョン3に向けて大きく変わった。マニュアルもきちんと本の形式になった。中身もリファレンスマニュアルとユーザーズマニュアルに分かれ，読みやすいものになっている。

同時に，トランスピュータ版もバージョンを3に上げた。標準システム版もトランスピュータ版もコンパチブルなので，今回は主にトランスピュータ版を使ってみた(だって速いほうがいいんだもん)。ボードを期限つきで借りたせいか貧乏性が出て，24時間フル稼働させてしまった。

レイトレーシングとはなにか

さて，レイトレーシングといわれてもわからない人のために若干の解説をしておこう。

まず，空間上に3次元図形を描く。といってもいきなり複雑なものを描こうとせず，簡単な方程式で表せるものを組み合わせることが多い。次に視点から画面の1ドットごとに飛ばした視線と物体の交点を求める。そこから光源まで光を飛ばし，その点の輝度を決定する。これを画面の全部の点に対して繰り返す。といった数学的方法で絵を描こうというわけだ。

こうすると「いかにもCG」という絵が描ける。計算を多用するだけあってパソコンレベルでは重い処理になるのだが，C-TRACEは32ビットのCG専用マシンと同等のレイトレーサをX68000上に持ってきたという素性の持ち主。時間さえかければ本格的CG画像ができるとあって，発表時のインパクトは大きかった。

今回はボクセル分割だ

レイトレーシングの描画は遅いといわれる。確かに3次元CGのアルゴリズムのなかではもっとも遅い部類に入るであろう。それでもレイトレーシングがすたれないのは，計算時間に見合うだけの作品ができあがるからなのである(異論がある方もいるだろうが)。が，綺麗だから遅くても我慢しよう，というわけにはいかないのがユーザーの偽らざる心境であり，高速化があれこれ試みられている。

C-TRACEは約1年前にハードウェアでの高速化を図ってトランスピュータ版をリリースした。計算を専門に行うトランスピュータボードを本体のスロットに差し込み，レイトレースの作業はすべてそちらに任せる。X68000は計算結果をもらって描画する部分などを受け持つだけである。

トランスピュータは高速化に絶大な力を発揮する。気軽に購入できる価格ではないが(X68000本体は軽く買えてしまう)，時は金なりともいうから，本格的にレイトレーシングをしようという方なら初期投資もすぐに回収できるだろう。

これに対し，ソフトウェアによる高速化のなかで現在もっとも効果を上げていると思われるのが，今回C-TRACEが採用したボクセル分割である。ボクセル分割は1年前にサイクロンExpressαで採用され大きな効果をあげている。ボクセル分割が登場するに至った構図は以下のとおり。

レイトレーシングは，画面上のピクセル1つひとつについて視線を発生させ，視線と最初にぶつかるプリミティブの色を画面にプロットして画像を作り上げていくアルゴリズムである。

↓

レイトレーシングの処理時間の大部分は，視線とプリミティブの交点計算に費やされている。

↓

しかもその大部分は無駄である。なぜなら，明らかに視線と交差しないプリミティブについても律儀に交点計算を行っているからである。

↓

これでは物体の数が増えれば増えるほど計算時間は増えていく。

↓

それでは，あさっての方向を向いている視線の計算を省くにはどうしたらいいか。

↓

視野を大まかなブロックに分割して，各プリミティブがどのブロックに入っているか調べておく。そして視線と交差するブロックに入っているプリミティブとの交点だけを計算する。

↓

こうすれば，物体の数が増えても計算時間が増えるのを抑えることができる。

＊　＊　＊

いつでも理屈どおりうまくいくわけではないが，このボクセル分割はある局面では非常に効果的である。プリミティブの数が多いと特においしい。

ボクセル分割はレイトレースの前処理として動作する。C-TRACEに描画コマンド(run)を与えると，最初の数秒～数分をボクセル分割に使い，それから描画を始める。たとえは悪いが，コンパイラ言語で実行の前にコンパイルが必要なのと同じようなものである。それでは実際に動かしてみることにしよう。

見よ！　ボクセル分割の威力（見えないって)。物体の数を多くしていってもゆっくりとしか遅くならない（？）のが快感。プリミティブの数に対する感覚が麻痺してしまいそうになる。

サンプルは今年の１月号のC言語特集で再帰呼び出しの応用例で作った木である。そのときよりも再帰レベルを深くして，枝の数を増やした（プリミティブの数は1366個)。これをトランスピュータを使って全画面（512×512ドット）に描画させてみた。これは従来では暴挙以外の何者でもなかったのだが……。

たったの30分。嘘ではない。アンチエリアシングを最高レベルでかけてみたが，それでも５時間。これを長いと思うなかれ。

いまのように反射も屈折もマッピングもしない場合はボクセル分割の威力がいかんなく発揮される。これがガラスなどのように反射・屈折が多発する材料だった場合，ボクセル分割といえどもがっくりスピードが落ちる。こんなときは，トランスピュータに感謝するのみである。

ボクセル分割では描画時間がシーンの状況に左右されるが，トランスピュータを使えば無条件に何倍も速くなる。同じことを標準システムでやった日には，X68000がハングアップしているようにしか見えないかもしれない。

大幅に拡張されたデータ形式

ところで，C-TRACEは，初期バージョンから２年間変わらなかった形状データファイル（RDTファイル）のフォーマットを変えた。大きな変更は３つ。そのうちうれしいのが２つ，個人的には歓迎したくないのがひとつ。歓迎したくないほうのひとつからいってみよう。

これがトランスピュータボードだ

ボクセル分割の導入に伴ってだろう（僕の推測だが)，使えるプリミティブが減った。具体的には，平面と双曲面（楕円体の拡張で，サイズ指定のところで負数を入れると使えた）が使えなくなっている。どこまでも続く図形というものは，存在範囲が限定できないので空間分割が命のボクセルと相性が悪いのであろう。

双曲面が使えないのではあまりに具合が悪いという判断からか，一葉双曲面（つづみのような形）だけは円柱の拡張という形で残された。つまり円柱の胴体を細らせると一葉双曲面なのである。前と違って蓋も底もあるので，ボクセルともなじむ。

まとめると，

・平面はなくなった。

・楕円体の拡張としての双曲面はなくなった。

・新たに円柱が拡張され，円柱の側面の形を変えられるようになった。そのなかに一葉双曲面も入っている。またそのために，円柱のサイズを指定するパラメータがひとつ増えている。

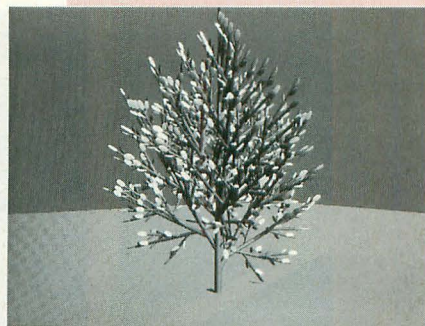
というわけで，従来のデータをそのまま読ませることはできなくなっている。さっきの再帰木にしても以前の記事のままではだめで，円柱のサイズ指定の部分を書き直したものである（蛇足だが，地面は平面ではなく直方体だったので変更は不要だった)。

なお，コマンド名のつづりが一部変わっている（たとえばアトリビュート指定がratt→ratなど）が，これは従来どおりの書き方でも読んでくれるようだ。

＊

あとのうれしい２つというのは，純粋な拡張である。従来どおりに書いてあっても通す，つまり指定しなかったのと同じ扱いになる。改めて移植することを考えなくてもよい。以後制作する作品に取り入れていけばいいというのが便利である。

ひとつはツリー構造。C-TRACEでは「クラスタ」と呼んでいる。サイクロンに対する最大の弱みといってもよかった部分がこれで解消されたことになる。

一般的なレイトレーシングでは，用意されているプリミティブの種類は少ない。球や直方体，それに円柱など，ごく簡単な形に限られる。それを積み木のように組み合わせていく。が，それだけで表現できる物体

これがたった30分でできる

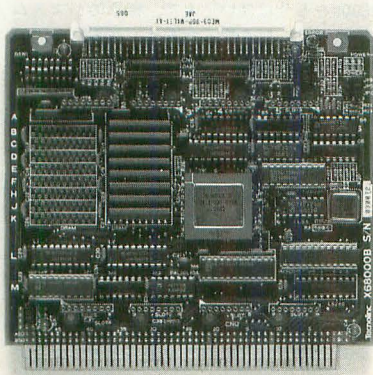
の形など知れたものだから，論理演算を使って物体をくりぬいたり，切り取ったりする。こうして一群のプリミティブからひとつの物体を構成する。問題はこの先だ。

C-TRACEのバージョン２までは，上のようにして作った物体をまとめて動かす手段がなかった。一群のプリミティブだからいっしょに動かせるというのが自然な流れだし，動かせないというのは非常に具合が悪いはずなのだが，動かせなかった。

そこで，プリミティブを１つひとつ動かしていくことを余儀なくされていたのだ。これはつまらないミスのもとだし，物体をあちらこちら動かしたいときにも苦労が絶えない。

たとえばグラスを作ったとする。置いてある場所が気にいらない。x方向に50動かしたい。グラスを構成するプリミティブを１個１個動かす。描画してみる。あれ，足が元の場所に取り残されている。しまった動かし忘れた。再度テキストエディタに戻って足を動かす。もう一度描画。グラスはちゃんと動いたが，この場所も気にいらない。z方向に30動かそうか。ああ，また同じ仕事の繰り返しだ。これでは誰だって嫌になる。

そこでクラスタの導入である。使い方は簡単で，グラスを定義した部分を「cls～clsend」で囲めばよい。PASCALの「begin～end」のようなものだ。clsの直後に動かすなり回転するなりの命令をつけるだけで，クラスタ全体が，つまりグラスが動くのである。さらにうれしいことに，マッピングの情報もクラスタの中に入れられる。クラスタを動かせば当然マッピングもいっしょになって動く。これは便利である。部品のライブラリ化が可能になったのだから。

ツリー構造というからには，ネスティングも可能である。相当複雑な表現も簡単にできることだろう。たとえば多関節キャラクタなんてどうだろう。クラスタは，アニメーションに利用することを多分に意識し

テーブルの上の光景。描画が速いと遊び心も広がる

ているのかもしれない。

うれしいこと2つめ，属性マッピング。サイクロンでいうアトリビュートマッピングができるようになった。いままではカラーマッピングとバンプマッピングだけだったのだが，今回は反射率・ハイライト・透明度などのマッピングまで可能になった。今回はうまい使い道を思いつかなかったが，それでもこっそりと使っている。これ見よがしに使うなんて感心しないからね，といいわけをしておこう。

モデリング環境は悪くなった?

順序が逆のような気もするが，モデリングのお話。といっても特別なソフトがあるわけではない，むしろなくなったというべきか。

C-TRACE付属のモデラーSPED.Xが，バージョン3のシステムから姿を消している。あのモデラーは，基本の部分がとてもよくできていたのに，残念である。

確かに扱える物体の数が少ない，論理演算をサポートしていない，プリミティブの追加はできても削除ができない，などの不満は数多くあった。しかし，操作性は上々，画面構成も小綺麗にまとまっていてわかりやすかった。レイトレース前に構図の確認ができたのだが，これからは出たとこ勝負，描画するまでわからない。今回はトランスピュータを使っていたのでまだよかったが，これを標準システムですることを考えると……。

データファイルのフォーマットが大幅に変わったので，それに追従するよりはモデラーを切り捨てる道を選択したのだろうか。まあツリー構造をサポートするのは大仕事には違いない。

そういうわけなので，自分にあったテキストエディタを見つけることが最重要課題となる。ツリー構造も導入されたことだし，編集機能が強力なエディタがほしいところ。よく手になじんだエディタは，半端なモデラーよりは確実に使える。

しかし，だからこそ，半端でないモデラーを出してほしかったのに，よりによって割愛とは……。

レイトレ作画講座もどきとサンプル

僕のような者が作画講座など開くのはおこがましい限りだが，はからずもサンプルの作り方がそれっぽくなってしまった。

先月はランダムフラクタルで雲を作ったが，これをさっそくマッピングに利用してみた。「箱庭(これぞ文字どおりのマッピング?)」では，カラーマッピングとバンプマッピングを同じ形にしてある。先月のランダムフラクタルに再現性があることを利用している。

ときには背景にも模様がほしくなる。しかしうまく立ち回らないとえらく間抜けなことになる。たとえば後ろのほうにでっかい板を立て，それに背景を貼りつける。そこで構図が気に入らなくなって，別の角度から描画してみる。すると，おお，背景が途中で切れて，その後ろが見えてしまっている。世界の果てでは海が滝になって流れ落ちているという話を笑えないではないか。背景の端が見えるのはそれくらいみっともない。

反射・屈折がからむともっと厄介だ。なにしろ反射光はどこからくるかわからない。ぱっと見にはうまくいっていても，映り込みの中に背景の板の形が見えていたりするのである。よく映画の特殊効果の種明かしをするメイキング物があるが，あれは映画と別にあるからいいのであって，映画の中でネタがばれてしまうような作り方はしてはいけない。ましてやレイトレーシングは1枚絵である(基本的には)。穴のあくほど見つめられる。ごまかしはきかない。

それでもトランスピュータがほしい

トランスピュータボードは安いとはいえない。もちろん，その性能を考えるとおつりが返ってくるし，今回は大幅な値下げまで行っている。しかし，まだまだ一般庶民がためらいもなくぽんと買える値段ではない。しかも用途は特殊である。

トランスピュータは，それ自体1個のCPUであるから，Cコンパイラだってある。並列処理を考えた設計だから，増設することでいくらでも速くなる。使い道はいくらでもあるのだが，現時点ではレイトレーシングに限られていると思っていいだろう。当然，そんなもの買ってなんになる?　と思う向きはひとりや2人ではあるまい。

ここにひとりのX68000ユーザーがいる。彼があるとき「コンピュータグラフィックをしよう」と思い立ったとしよう。彼に求められるものはなにか。財力?　いやいやそんなことはない。X68000は標準装備だけで相当なことができるマシンだ。彼に必要なのは，努力とセンスだけである（どちらもお金で買えないものだけに，かえってつらい)。

かつて，X1がスーパーインポーズやPCGやジョイスティックポートを標準装備して登場したときも悪口は絶えないものであった。いま，それらに疑問を抱く者がいるであろうか。ジョイスティックポートにつながったのはジョイスティックだけではないことも思い出してもらいたい。

世の中には僕らが考えているよりもはるかにたくさんの才人が埋もれているかもしれない。もしもその人たちが，音楽でもグラフィックでもいいのだが，なにか特別なことをしようと考えたとする。そしてその人の使っているマシンがまったくのダルマさんだったらどうだろうか。「音楽をしてみたいんですが」「FM音源ボードを買ってください」「グラフィックをやってみたいんですが」「××社のフレームバッファを買ってください」。……拡張，拡張，拡張。なにをするにも拡張というのは，「無駄な機能はコストを押し上げる」という考え方からいえばたいへん合理的ではあるが，へたをすると優れた才能の芽を摘むことにもなる。

初めから音楽が目的だ，とか，CGに憧れてコンピュータを始めた，とかいう人ならば，その方面に惜しみなくお金をつかうことができる。でも，コンピュータを始めてからそういったものに触れる人のほうが世の中には圧倒的に多いのである。好きならば求めればいい，しかし出会いがなければ嫌いにすらなれない。X68000は，その出会いのチャンスを豊富に与えてくれる希有のマシンである。

X68000を十分に使い込んで，標準のシステム構成に飽き足らなくなった人は，さらに専門的なシステムを構築していけばよい。ここから先は，コストが上がっていっても少しも構わない。より強力になったシステムは，注ぎ込んだ費用以上の働きをしてくれるだろう。人によってはそれがMIDIだったり（MIDIはX68000にとってそれほど特殊ではなくなっているが)，フレームバッファやトランスピュータだったりするわけである。

その道の素人でも優しく迎え入れ，達人になったあとも要求に答えていける，そんな包容力を持ったマシンを持てることは幸せである。以上，システムアップに対する僕の考えである。

C-TRACEのディスクにはサンプルがついてくる。例のステゴちゃんのデータも入っていた。そこから解決法を頂戴した。よくできたサンプルを読むと勉強になる。

巨大な球を作ってその裏側にマップを貼る。基本はこれだけ。ただ，これだけだといろいろ不都合を生じる可能性もあるので注意されたい。

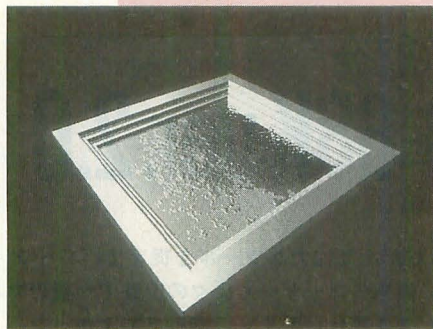
まず光源。球の外側からくる光は球の内側には届かないかもしれない。ひょっとしたら背景の球の内側が真っ暗になるかもしれない。すると，背景はおろか肝心の被写体も真っ黒になる。

対策その1。球の一部にだけマッピングする（例：「時計」の青空），これは映り込みでマップの形が見えないように注意する必要がある。球面マッピングは，マッピング範囲を角度で指定するので設定はしやすい。

対策その2。球の全面にマッピングしておき，その内側に点光源を置く（「シャボン玉」）。

余談になるが，ここで疑問を持った方も多かろう。「シャボン玉」は背景にマッピングしてあるようには見えないだろうから。シャボン玉の表面をご覧いただきたい。この油膜の虹色模様は，背景が映り込んでいるのだ。これは僕ではなく，中野氏の提案による技である（感謝）。球面の内側にフラクタルで作った虹色をマッピングしておく。点光源を背後に置く。背景のうち，後半分だけが明るくなる。それが玉（正体はただの無屈折のガラス玉）の表面に映り込む。種を明かせばそれだけのことだが，けっこう雰囲気は出ている。ごまかしだって？いやだなあ人聞きの悪い，せめて高等テクニックといってくださいよ。

ほかにも「球の表面に反射率マッピングする」という，よりそれっぽいごまかし，もとい，高等テクニックもある。ま，真剣にシミュレートしようと思ったら光の干渉を考えに入れるべきである。というか，実際にレイトレーシングでそういう試みはなされている。もっともこれはオリジナルにレンダラを組まねばならないが。

脱線したが，対策その3。背景色（bgcコマンドで与える。バック・グラウンド・カラーの略であろうか）を明るめの白にしておいて，球面には透明率マッピングを施す（例：「再帰木その2」の夕焼けっぽい空）。この方法ではムラのない背景が得られる（前2つではムラが出る可能性がある。それを利用するのもテクニックのうちであろう）。

バンプマッピング

これもカラーマッピング

最後に一般論。

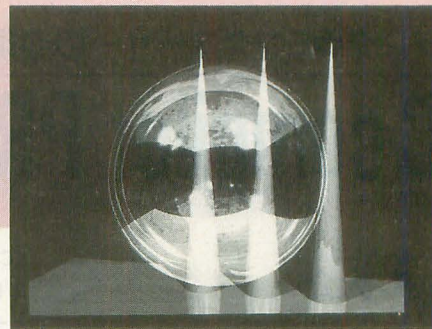
綺麗な絵は作るのが難しい。特にライティング（照明）は重要で，これの指定いかんでシーンがシラけてしまったり，妙にリアルになったりする。

お金があるっていいことね

レイトレーシング一般にいえることだが，少し真面目にやろうと思うと途端に膨大な資源を要求する。

まずハードディスクはもはや必需品。マッピング用イメージファイルには，1枚200Kバイトなんてのもざら。C言語を使って自動生成した「再帰木」の形状データファイルのサイズもそう。指数関数的に増大するので，300Kバイトくらいは軽くいく。レベルをちょっと上げたらファイルサイズが3Mバイトいった。これはさすがにトランスピュータのほうでメモリオーバーを起こしたので読み込めなかったが，フロッピーに入りきらないテキストデータを作ったこと自体ちょっとしたスリルであった。

出力される画像ファイルも全画面で700Kバイト以上。圧縮もかかっていない。これでアニメーションなんてしようものなら，ハードディスクでもパンクしてしまう。光磁気ディスクの時代はそこまで来ている。

トランスピュータもぜひともほしいところ。いまならさらに値下がりしてお得だ。大金持ちの人は，追加ボードでも載せて速度を一気に2倍にすることもできる（レイトレーシングはアルゴリズムの性格上，速度がプロセッサ数に比例する）。描画が遅いために生じるストレスから解放されないと，感性の活躍もなにもあったものではない。

今回はけっこう楽しかった。トランスピュータにボクセル分割で鬼に金棒。今回のサンプルのほとんどは，解像度を最大にしても30分で終わってしまった。テスト描画に至っては数十秒である。

「時計」にしても最初はただの1枚板だったのが，バンプをつけ，針をつけ，ガラス板をつけ，四隅にピンをつけ，気の向くままに造形ができて楽しかった（そのあとで背景を青空にしたが，これは少々突飛だったかな）。ともあれ，造り込むとなんとなくそれらしく見えてくるから不思議ではある。少々派手にしたって大丈夫，トランスピュータにはメモリが4Mバイト入っている（拡張すれば8Mバイトまで可能だそうだ）。マッピングもけちってはいけない。

なんだかトランスピュータ礼賛になってしまったが，トランスピュータがここまでおいしくなったのもバージョン3だからだということを忘れてはならない。ボクセル分割万歳。さあ，次はメタボールだ。

アニメーションフレーマー

レイトレーシングでアニメーションなんてできるのはよほど資金力のあるところだろうと思うが，理論上は誰でもアニメーションを作ることができる。そこで問題になるのが動きの設定。複雑な動作を指定するのはけっこう大変な作業である。なにが大変かといって，1秒に30フレームもの指定をするのが大変なのである。

こんなとき，たとえば1秒につき5フレームずつ指定しておいて残りの25フレームは中割りで作らせるということができれば，作業の効率がぐっと上がることが期待できる。

今回付属のアニメーションフレーマーは，C-TRACEで指定できるすべてのパラメータを直線またはスプラインで補間する。プリミティブのサイズ，座標，色はもちろんのこと，構図も滑らかに変えていくこともできる。カメラワークの勉強が必要だな。

といっても正体はフィルタのようなもの。アニメーションフレーマーはC-TRACE本体とは独立したアプリケーションで，RDTまたはKEYファイルのパラメータを細かく書き換えながら出力するだけのものである。ただ，レイトレーサ自体を子プロセスとして呼び出すことができるので，全フレームを自動生成してくれるのがうれしい。

C-TRACEが面倒を見るのはレンダリングまでで，そこから先の動画像の再生についてはサポートがないようである。これについては，そろそろ出てきそうな「うごくZO」がなんとかしてくれそうな気配である。

プログラミングユーザーの必須アイテム

C compiler PRO-68K ver.2.0

Izumi Daisuke
泉 大介

XCがいちだんと強力なツールを従えてパワーアップしてきました。従来のユーザーも少々の出費で大量のマニュアルごとサービスを受けられます。持っててよかった。

待望久しいC compiler PRO-68K（通称，XC）のver.2.0がついに編集部に届きました。長い道程でしたが，これは！ と思わせる新機能を搭載しての登場です。

新機能の第1点はソースコードデバッガが付いたことです。これを使うとCのプログラムを1行ずつ実行しながら変数の値を確かめることができるようになります。

第2点はANSI規格をより取り入れた表記をサポートするようになったことです。

第3点はライブラリ環境が整備されたということです。これまで複数のファイルを単にまとめた～.Aというファイルをライブラリ代わりに使ってきましたが，今回専用のファイル形式が用意されました。

そして第4点は，生成されるコードの質が少し向上したことです。

怒濤のマニュアル攻撃

まずは製品構成から始めましょう。C compiler PRO-68K ver.2.0のパッケージに含まれるものは，

1) XCシステムディスク1
2) XCシステムディスク2
3) XCライブラリディスク
4) Cユーザーズマニュアル(169P)
5) Cリファレンスマニュアル(136P)
6) CライブラリマニュアルVOL.1(621P)
7) CライブラリマニュアルVOL.2(469P)
8) ソースコードデバッガマニュアル(126P)
9) アセンブラマニュアル(310P)
10) プログラマーズマニュアル(754P)
11) 登録カード

の合計11アイテムです。マニュアル類の後ろの括弧に入れたのはページ数で，ver.1で究極のコピープロテクトと呼ばれた怒濤のマニュアル攻撃はさらに磨きをかけられています。マニュアル総量は2585ページに達し，その重量はなんと5.5kg。ライブラリマニュアルは2冊に分けられ，X68000とHuman68kに関するあらゆる情報が詰め込まれています。

気になる価格ですが，ver.1より5,000円アップの44,800円に決まりました。ver.1ユーザーには差額＋1万円程度でバージョンアップサービスしたいとのことです。登録カードを返送している方には，もうバージョンアップの案内が届いている頃ではないでしょうか。

さて，ver.1ではシステムディスクは1枚しかなく，買ってきたシステムディスクを起動するだけでCコンパイラを使えるようになっていました。ソースコードデバッガなどの機能が追加されたver.2.0ではシステムディスクは2枚構成となり，起動するとインストーラが自動的に立ち上がります。これは2枚のシステムディスクに振り分けて収められた種々のファイルを，目的別にディレクトリを作成して分類整理してくれるものです。

フロッピーディスクへのインストールとハードディスクへのインストールの2つが用意され，いずれも画面の指示にしたがって操作していくだけで簡単にインストールできるようになっています。フロッピーディスクへのインストールでは起動用と実行用の2枚のフロッピーディスクを作成します。

ここで残念なお知らせがあります。こうして作成した起動用ディスクでは，1Mバイトしかメモリを搭載していないマシンでCコンパイラを使えません。ver.1ではCコンパイラは（BC.X），CC.X，CCP，CC0，CC1，CC2，そしてAS.Xの7つのプログラムに分かれていました。これらが順次起動されながらコンパイルが進んでいくようになっていたのですが，今回のバージョンアップで7つのファイルはCC.Xという1つのプログラムにまとめられたのです。この結果CC.Xは約400Kバイトという巨大なプログラムになっています。

このように変更された理由は，複数のファイルから成るCのプログラムをコンパイルする際に，何度もCCP，CC0，CC1……を読み込む手間を省くためだと思われます。いろいろと試した結果コンパイルには700Kバイトほどのフリーエリアが必要なようですので，作成されたディスクのCONFIG.SYSからASK68KやOPM/PCMドライバ，それにヒストリドライバのすべてを削除すれば1MバイトのX68000でも使えるようになります。

ソースコードデバッガ

ver.2.0のもっとも大きな特長は，ソースコードデバッガが付属しているということです。Cで作成したプログラムはコンパイラによりマシン語に変換されます。プログラムの実行はマシン語ですから高速で，これがC言語が普及する要因のひとつともなりました。ところがプログラムの動作が変なときには，BASICのように途中で止めて変数の値を確認するといったテストができないというデメリットも併せ持っています。このような場合，ver.1では，

1) Cのプログラムをじっと睨む
2) プログラムのところどころに変数を表示する命令を入れて再びコンパイルし，表示される値を見ながらおかしなところを探し出す
3) マシン語を自分で調べておかしなところを探す

といった方法で原因をつきとめるしかありませんでした。特に3)の方法は自分の思い込みや勘違いを発見するのに有効な方法なのですが，マシン語レベルの知識とCコンパイラがどのようにマシン語へ変換するのかを知っていなければなりません。一般のC言語ユーザーにはとてもこんなことは無理な相談で，1)や2)の方法で地道にバグをつきとめるのが一般的な解決策だったといえるでしょう。

ソースコードデバッガはこのような暗黒のデバッグ時代に光明をもたらす画期的なデバッガです。C言語で書いたプログラムを見ながら自由にプログラムを動かしたり止めたりでき，変数の内容を表示させて確かめることが可能なのです。マシン語の知識は必要ありません。表示したい変数名を指定し，止めたいところでマウスをクリック。これだけで勘と知識と忍耐を総動員して立ち向かっていたバグと対決できるわけです。

ver.2.0付属のソースコードデバッガSCD.Xはマウスオペレーションを基本とした簡便な操作が特長です。画面は大きく

５つに分かれており，いちばん上がメニューバー，次がソースプログラム（この場合はC言語）を表示するためのウィンドウになっています。左上にある上下の三角印は表示されているプログラムのスクロール用です。３番目のウィンドウはMC68000のレジスタやCの変数の内容を表示するためのもの。４番目がコマンドを入力するためのウィンドウで，コマンド投入ウィンドウと名付けられています。

いたずらにマウスに固執することなく，必要に応じてマウスとキーボードの両方を使える設計には好感が持てます。いちばん下のウィンドウ（画面下の黒い部分）はアプリケーションが通常使っているテキスト画面が一部分だけ見えているところです。SCD.Xはマウスカーソルと同じ画面を使っていますので，通常のプログラムの画面出力にはまったく影響を与えません。

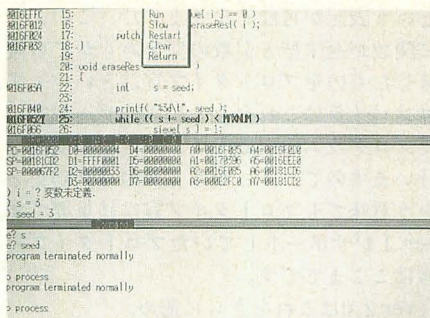
もっとも簡単なデバッグ方法はメニューのTraceあるいはStepでマウスをクリックすることです。クリックするたびにC言語のプログラムが１行ずつ実行されていき，実行中の行が網掛けで表示されます。プログラムを１行ずつ追いかけながら動きを確認できますので，実行されないif文のチェックなどに役立ちます。Traceは呼び出している関数の中まで１行実行する点がStepと異なっています。

変数の表示はwatchを選択しプルダウンメニューからsetを選ぶだけと簡単。変数名入力用のウィンドウが開きますので，見たい変数名を入力します。これで２番目のウィンドウに変数名とその値が表示されます。「変数未定義」と表示されたら，それは指定した変数が現在実行中の場所からは参照できないという意味です。

先ほどのステップ実行と併せれば，初心者の方の格好の入門教材となるでしょう。変数の内容を16進数で表示したり，文字列として表示できますので，適宜使い分けるのがいいでしょう。ただ，変数に配列名を指定すると，

sieve = Array：アドレス

と表示されるだけで内容までは表示してくれません。もっとも1000個もの要素をもつ配列の内容を表示されても困るだけのような気がしないでもないのですが。この場合は，たとえば「sieve [i]」を表示するというように，添え字（変数で指定してもかまわない）を付けて表示させることになります。

初心者のための入門教材という意味では，Execメニューの中のSlowも捨てたものではありません。これは自動Traceともいうべき機能で，１行ずつ順に実行していってくれます。自分の作ったプログラムが動いていく様子を見ているのはなかなか面白いものです。

マウスで指定した行数（25行）まで実行させる

網掛けになっている行を実行中はSCD.Xの画面が一時的にOFFになります。このためSlow実行では画面が点滅するような格好になり目の衛生上よくありません。こんなときに便利に使えるのがいちばん下のテキスト画面です。

黒いところでマウスの左ボタンを押すとUser Screenと書かれたバーが現れます。そのままドラッグするとサイズは自由に変更でき，テキスト画面を常に表示しながらプログラムを追いかけることができるようになるのですが，もうひとつ美味しい機能が付加されます。SCD.Xの画面が消えなくなるのです。チラチラが気になる方は，１行分だけテキスト画面を取っておくことをお勧めします。

１行ずつの実行ではなく一気に数行実行したいというときには，止めたい行でマウスの右ボタンを押すだけでOKです。指定した行までのプログラムが実行され，網掛け表示されて中断します。

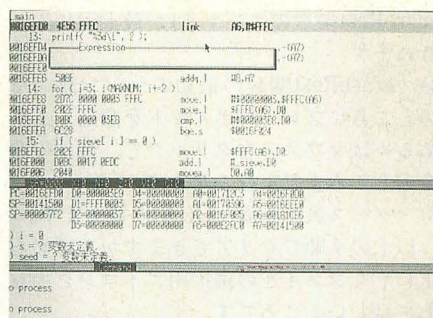
同様の機能にブレイクポイントの設定があります。こちらはマウスの左ボタンで設定し，Runを選んでプログラムを走らせれば設定した行で中断します。いずれも大きなプログラムで，問題があると思われる関数が呼び出されるまでは普通に実行したい，というときに有効な方法です。目的の場所までは高速に実行させ，怪しそうなところに近づいたらステップ実行で原因を探るという使い方をします。

●ちょっと高度なデバッグ方法

実行を中断する行を指定する，指定した行まで実行するといった前述の方法のほかに，条件が成立したら実行を中断するという方法があります。これはTrace, Step, Slow 実行時しか有効ではないのですが，メニューのWatchからWatchPtを選んで，

i == 5

のように指定しておけば，iが５になったときに自動的に停止する機能です。

さらにすごいのがTracePt。こちらはメモリの内容が書き換えられたら停止するという優れものです。プログラムのどこかが配列の内容を壊しているとき，あるいは配列の範囲を越えてデータを書き込んでいるときには，見張るメモリを指定してSlow実行させておけばOKです。血眼になって探していたバグが，寝そべって本でも読んでいる間に探し出されます。

ソースリストを参照する

SCD.Xは従来のデバッガDB.Xをソースコードデバッガへと進化させたものです。したがって，DB.Xで使っていたデバッグテクニックをそのまま利用することが可能です。これらはコマンド投入ウィンドウでコマンドを入力して使うことになり，DB.Xと同じような操作環境になります。先ほど配列の内容は表示されないと書きましたが，メモリダンプ命令を使って

d. _sieve

とすれば表示できるわけです。XCはシンボルの前にアンダーバーを付けて使いますので，配列sieveを見たければ「_sieve」で指定します。

実は，SCD.Xを使っているうちに大切な機能が抜けていることに気づきました。一般にCのプログラムは機能ごとに複数のファイルに分割して作成されます。SCD.Xでもこれに対応し，実行が別のソースファイルに移ると自動的にソースファイルを切り替えて表示してくれるようになっています。ところが，このソースファイルの切り替えを任意に行うことができないようなのです。

「プログラムの動作を見ていると，どうも５つ目のファイルのあの関数がおかしそうだ」と思っても，５つ目のファイルを表示できないのですからブレイクポイントを設定できません。トレースなどを繰り返して目的の関数へ辿り着かなければならないわけです。これは致命的だと思ったのですが，とりあえず抜け道を発見したのでここに記しておきます。

1)　まずFindメニューからSymbolを選びます。これはシンボルの検索を行うメニューです。

2)　ここで「_eraseRest」のように，ブレイクポイントを設定したい関数を「シンボル」で指定して探します。

3)　ソース表示画面が逆アセンブルリスト表示になり，検索したシンボル（つまり

eraseRest関数のエントリ)が最上行に表示されます。
4) eraseRest関数の最初のマシン語命令のところにブレイクポイントを設定します。あるいはマウスの右クリックで，最初の命令のところまで実行させます。
5) ソース表示画面をCに切り替えます。

以上の手順でクリアできますが，画面右上にでもファイルの選択用アイコンを用意してほしいところです。

よりANSI規格を取り込んで

数年前には，C言語の文献といえばカーニハンとリッチーのプログラミング言語Cしかないという状況でした。開発者によって著されたこの本はCという言語の紹介から処理系の詳しい内容まで多岐に渡っていましたが，どちらかというとCの精神論的な側面が強く処理系制作者がよりどころとするには厳密に定義されていない部分や処理系制作者に任されている部分も少なからずありました。それぞれの処理系はこの本を元にしながらも，少しずつ違ったものが世に送り出されたのです。こうした事態の収拾に乗り出したのがANSIで，以後ANSIが提出した標準Cの草案を満たすべくCコンパイラが作成されるようになりました。

ver.1でもこの草案のいくつかは取り入れられていますが，ver.2.0はさらにANSI準拠の性格が強まったCコンパイラになっています。そのいくつかを具体的例を挙げながら紹介していきましょう。

●関数定義方法の追加

ver.1では関数引数部は，

```
void eraseRest (seed)
int seed;
```

のように書かなければなりませんでした。ver.2.0ではこれを，

```
void eraseRest (int seed)
```

と表記できるようになっています。引数名だけでなくその型も同時に宣言できるようになったのです。

これはプロトタイプ宣言とも密接な関係があります。古いCではint以外の型のデータを返す関数は，あらかじめ，

```
double copy ( ) ;
```

のようにその型を宣言しておく必要がありました。続いて引数の型も同時に宣言しておけるようになり，

```
double copy (double, double) ;
```

という表記が可能となりました。このように関数が返す型と引数の型をあらかじめ宣言したものをプロトタイプ宣言といいます。コンパイラはこれをもとにユーザーが書いたプログラムの引数のチェックや型変換を行いますので，うっかりミスのバグ防止という意味でもプロトタイプ宣言は重要です。ver.1がサポートしていたプロトタイプ宣言はここまでです。

ver.2.0はこれをさらに進め，

```
double copy (double from,
    double to) ;
```

と，変数名までプロトタイプ宣言に入れることが可能となっています。変数の意味がわかりやすいだけでなく，関数定義行をコピーしてファイルの先頭に持ってくるだけでプロトタイプ宣言行を作ることができるのは便利です。

ANSIではprintf関数のように不定個の引数を受け取る関数を作る方法をサポートしていますが，これも今回のバージョンアップで取り入れられました。

●構造体・共用体の柔軟な扱い

XCはver.1のときから構造体・共用体を単純な変数と同じように扱うことができました。構造体どうしの代入をサポートしており，また関数の引数・戻り値としても使用可能でした。PC-9800シリーズ用のCコンパイラがこういった機能をサポートし始めようとしていた時期に，ver.1ではすでに実用化されていたわけです。ただ，関数の引数として使える構造体は1つだけで，2つ以上の構造体を渡そうとするとアドレス計算を間違えるというオマケがついてはいましたが。

ver.2.0になって複数の構造体を引数として渡した際のバグが取り除かれています。

●ライブラリの拡充

ANSIはその草案の中でどのコンピュータのCコンパイラでも使えるライブラリ関数を定義しており，ANSI標準ライブラリと呼ばれています。標準ライブラリで提供されている機能はマシンを問わず実行できることが期待されるわけで，XCでも今後とも拡充が望まれるところです。

ver.2.0はハードウェアサポートライブラリに加え，ver.1でサポートしていた関数のいくつかを標準ライブラリの仕様に沿う方向で変更しています。もっとも影響があるのではないかと思われるのはsetjmp.hとsetjmp, longjmp関連の処理です。これまで両関数の引数はjmp_bufへのポインタ型でしたが，これがjmp_buf型へと改められました。頻繁に移植をしていた方はsetjmp.hを書き換えてあるでしょうから問題ないとして，提供されていたsetjmp.hをそのまま使ってプログラムを作成していた方はプログラムを書き換えないとコンパイルできません。ご注意ください。

リスト1

```
REM LK.X VER2.0をGCCで使う

gcc -c -O sieve.c
lk sieve a:¥lib¥gnulib.a a:¥lib¥floatfnc.l
```

ライブラリの整備

これまでライブラリ代わりに使用してきた〜.Aファイルは複数のファイルを単純に結合し1つのファイルにまとめたものでした。ver.2.0ではライブラリは〜.Lという専用のファイルにまとめられています。ライブラリファイルを変更した理由は，これまで〜.A内に分散していたシンボル情報をファイルの先頭に集めてリンク速度の向上を図るためだそうです。これにあわせてLK.Xもバージョンアップされています。

ARコマンドで行っていたライブラリの管理はLIBコマンドに任されることになりました。ver.1ユーザーのために従来の〜.Aを〜.Lへ変換する機能がついていますので，自作ライブラリをver.2.0で使用するときにはこれで変換するといいでしょう。マニュアルではLK.X ver.2.0は〜.Aファイルもリンクできることになっているのですが，編集室に届いたバージョンではコプロセスで使うとエラーが出てしまいました。

CC.XやGCC.X (GNU C) はいくつかのコマンドをコプロセスで順番に起動しながら作業を進めていきます。したがってCC.Xで〜.A形式のライブラリを使用することはできません。本誌6月号の付録ディスクで配布したGCCは必ずGNULIB.AとCLIB.Aをリンクしますので，こちらでもそのままでは使用できません。GCCを使う場合は-cオプションを指定してリンクフェーズをカットし，次にLK.X ver.2.0でリンクするという使い方をすることになります。

＊

さて，コンパイラがはき出すコードの最適化処理なども気になるところですが，ここでは1000までの素数をエラトステネスのふるいで求める簡単なプログラムをコンパイルした結果を掲載しておきます。比較したのはver.1，ver.2.0，そしてGCCです。コンパイル条件はすべてオプティマイズオプションのみを指定しました。ver.1とver.2.0を比べると，共通文字列を削除している点，スタックに数値を積む場合にPEAを使っているほかはほとんど同じコードが生成されています。コードの質はver.2.0になって少し向上したというところでしょうか。

C compiler PRO-68K ver.2.0はX68000

でプログラミングを行うための総合開発セットという意味合いを持っています。MAKE.Xという強力なファイル保守ユーティリティもついてきます。実はあまりの多機能さに今回は紹介できませんでした。

また，FM音源関係ではOPMドライバが新しいバージョンとなり，追加BASIC関数としてMIDIへの対応もサポートされています。なお，登録されているMUSIC2.FNCが本誌で発表したものと同じファイル名となっているようですが，これらは追ってレポートすることにしましょう。

今回の目玉はなんといってもソースコードデバッガです。MicrosoftがCodeViewというソースコードデバッガを発売したとき，「神の作り給うたデバッガ」と評した記事がありましたが，この感覚を味わっていただけるのではないかと思います。

リスト2-a

```
==================  sieve.c  ==================
 1: #include <stdio.h>
 2:
 3: #define MAXNUM 1000
 4:
 5: void eraseRest( int );
 6:
 7: char sieve[1000];
 8:
 9: void main( argc, argv )
10: int     argc;
11: char    *argv[];
12: {
13:         int     i;
14:
15:         printf( "%3d¥t", 2 );
16:         for ( i=3; i<MAXNUM; i+=2 )
17:                 if ( sieve[ i ] == 0 )
18:                         eraseRest( i );
19:         putchar( '¥n' );
20: }
21:
22: void eraseRest( seed )
23: int     seed;
24: {
25:         int     s = seed;
26:
27:         printf( "%3d¥t", seed );
28:         while (( s += seed ) < MAXNUM )
29:                 sieve[ s ] = 1;
30: }
```

リスト2

2-b

```
==================  sieve.cc  ==================
 1: *  エラトステネスのふるい  XC VER1.01
 2:
 3:         include fefunc.h
 4:         .COMM   _sieve,1000
 5:         .XREF   __main
 6:         .XDEF   _main
 7:         .TEXT
 8: _main:
 9: __main:
10:         BRA     L1
11: L20001:
12:         MOVE.L  -4(A6),D0
13:         ADD.L   #_sieve,D0
14:         MOVE.L  D0,A0
15:         MOVE.B  (A0),D0
16:         EXT.W   D0
17:         EXT.L   D0
18:         TST.L   D0
19:         BNE     L8
20:         MOVE.L  -4(A6),-(SP)
21:         JSR     _eraseRest
22:         ADDQ.L  #4,SP
23: L8:
24: L7:
25:         ADDQ.L  #2,-4(A6)
26: L5:
27:         MOVE.L  -4(A6),D0
28:         CMP.L   #1000,D0
29:         BLT     L20001
30:         MOVE.L  #10,-(SP)
31:         JSR     _putchar
32:         ADDQ.L  #4,SP
33: L3:
34:         UNLK    A6
35:         RTS
36: L1:
37:         LINK    A6,#-4
38:         MOVE.L  #2,-(SP)
39:         MOVE.L  #L4,-(SP)
40:         JSR     _printf
41:         ADDQ.L  #8,SP
42:         MOVE.L  #3,-4(A6)
43:         BRA     L5
44:         .GLOBL  _eraseRest
45: _eraseRest:
46: __eraseRest:
47:         BRA     L9
48: L20003:
49:         MOVE.L  -4(A6),D0
50:         ADD.L   #_sieve,D0
51:         MOVE.L  D0,A0
52:         MOVE.L  #1,D0
53:         MOVE.B  D0,(A0)
54: L13:
55:         MOVE.L  8(A6),D0
56:         ADD.L   D0,-4(A6)
57:         MOVE.L  -4(A6),D0
58:         CMP.L   #1000,D0
59:         BLT     L20003
60: L11:
61:         UNLK    A6
62:         RTS
63: L9:
64:         LINK    A6,#-4
65:         MOVE.L  8(A6),-4(A6)
66:         MOVE.L  8(A6),-(SP)
67:         MOVE.L  #L12,-(SP)
68:         JSR     _printf
69:         ADDQ.L  #8,SP
70:         BRA     L13
71:         .DATA
72: L4:
73:         .DC.B   $25,$33,$64,$09,$00
74: L12:
75:         .DC.B   $25,$33,$64,$09,$00
76:         .EVEN
77:         .END
```

2-c

```
==================  sieve.cc2  ==================
 1: *  エラトステネスのふるい  XC VER2.0
 2:
 3:         INCLUDE fefunc.h
 4:         .GLOBL  _main
 5:         .GLOBL  _eraseRest
 6:         .XREF   __main
 7:         .TEXT
 8: _main:
 9:         LINK    A6,#-4
10:         PEA     2.W
11:         MOVE.L  #L4,-(SP)
12:         JSR     _printf
13:         ADDQ.L  #8,SP
14:         MOVE.L  #3,-4(A6)
15:         BRA     L5
16: L20001:
17:         MOVE.L  -4(A6),D0
18:         ADD.L   #_sieve,D0
19:         MOVE.L  D0,A0
20:         MOVE.B  (A0),D0
21:         EXT.W   D0
22:         EXT.L   D0
23:         TST.L   D0
24:         BNE     L7
25:         MOVE.L  -4(A6),-(SP)
26:         JSR     _eraseRest
27:         ADDQ.L  #4,SP
28: L7:
29:         ADDQ.L  #2,-4(A6)
30: L5:
31:         MOVE.L  -4(A6),D0
32:         CMP.L   #1000,D0
33:         BLT     L20001
34:         PEA     10.W
35:         JSR     _putchar
36:         ADDQ.L  #4,SP
37:         UNLK    A6
38:         RTS
39: _eraseRest:
40:         LINK    A6,#-4
41:         MOVE.L  8(A6),-4(A6)
42:         MOVE.L  8(A6),-(SP)
43:         MOVE.L  #L4,-(SP)
44:         JSR     _printf
45:         ADDQ.L  #8,SP
46:         BRA     L13
47: L20003:
48:         MOVE.L  -4(A6),D0
49:         ADD.L   #_sieve,D0
50:         MOVE.L  D0,A0
51:         MOVE.L  #1,D0
52:         MOVE.B  D0,(A0)
53: L13:
54:         MOVE.L  8(A6),D0
55:         ADD.L   D0,-4(A6)
56:         MOVE.L  -4(A6),D0
57:         CMP.L   #1000,D0
58:         BLT     L20003
59:         UNLK    A6
60:         RTS
61: L4:
62:         .DC.B   $25,$33,$64,$09,$00
63:         .DATA
64:         .EVEN
65:         .BSS
66:         .COMM   _sieve,1000
67:         .END
```

2-d

```
==================  sieve.gcc  ==================
 1: *  エラトステネスのふるい  GCC VER1.37
 2:
 3: * NO_APP
 4: gcc_compiled:
 5:         .text
 6: LC0:
 7:         .dc.b $25,$33,$64,$09,$00
 8:         .even
 9:         .xref __main
10:         .xdef _main
11: _main:
12:         link a6,#0
13:         movem.l d3/a3,-(sp)
14:         pea 2
15:         pea LC0
16:         jsr _printf
17:         moveq.l #3,d3
18:         addq.w #8,sp
19:         lea _sieve,a3
20: L6:
21:         tst.b (a3,d3.l)
22:         bne L4
23:         move.l d3,-(sp)
24:         jsr _eraseRest
25:         addq.w #4,sp
26: L4:
27:         addq.l #2,d3
28:         cmp.l #999,d3
29:         ble L6
30:         pea 10
31:         jsr _putchar
32:         movem.l -8(a6),d3/a3
33:         unlk a6
34:         rts
35:         .even
36:         .globl _eraseRest
37: _eraseRest:
38:         link a6,#0
39:         movem.l d3/d4,-(sp)
40:         move.l 8(a6),d3
41:         move.l d3,d4
42:         move.l d3,-(sp)
43:         pea LC0
44:         jsr _printf
45:         lea _sieve,a0
46:         bra L8
47: L10:
48:         move.b #1,(a0,d3.l)
49: L8:
50:         add.l d4,d3
51:         cmp.l #999,d3
52:         ble L10
53:         movem.l -8(a6),d3/d4
54:         unlk a6
55:         rts
56:         .data
57:         .bss
58:         .xdef _sieve
59:         .comm _sieve,1000
60:         .text
```

ようこそここへC言語

プログラムって何だろう

[第1回]

Nakamori Akira
中森　章

皆さんお待たせしました。いよいよC言語入門講座が始まります。入門といってももちろん中身は本格派。でも大丈夫，講師の中森先生には高度になりすぎないよう編集部からお願いしてあります。ゆっくりそしてじっくりと攻めていきましょう。

はじめに

祝一平氏のあとを受けて今月からC言語の連載を始めることになった中森章です。祝氏の連載はC言語の基礎よりもその応用面に力を入れていたため初心者にはつらいものがありました。この連載ではC言語の応用よりも基礎的な面に焦点を当てて，初心者が簡単なプログラムを作れるようになることを目標にしたいと思います。ここではC言語に関する予備知識は特に必要としません。ただ，別のプログラム言語（BASICでよい）をさわったことがあれば理解が少し早くなるかもしれません。

連載の進め方としては，プログラムの考え方（概念）から始めてそれを実現する例として実際のプログラムを提示する方法を（可能な限り）とりたいと思っています。これは「型」から入る従来の方法とは少々異なります。もちろん紋切り型の「型」を繰り返すうちにその本質に近づいていくという方法論は大切ですが，私は「型」の本質を知ったうえで「型」を繰り返しながらそれを復習していくほうが効果的だと思います。まあ，どちらにしてもできるだけ多くのプログラムを繰り返し作ってみることが必要なのは同じです。

使用するCコンパイラはXCを考えていますが，基本的にはどのCコンパイラでも動作すると思われます。一応Human68k上のXCとGCCでの動作は保証することにしましょう。

図1　プログラミングの一般的概念

図2　プログラムの実行段階の概念

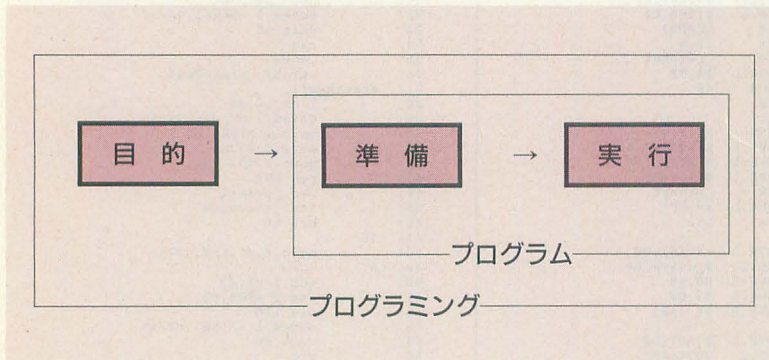
プログラムというもの

C言語に限らずプログラミングを行う前に確認しておかなければならないことがあります。プログラミングとはプログラムを書くことですが，そもそもプログラムとはなんなのでしょう。これがわからないうちはいくらプログラミング言語の文法を知っていてもプログラムを書けるようになりません（読むことはできるでしょうけど）。そこでまず「プログラムとは何か」について考えてみましょう。

●原因と結果（入力と出力）

私たちがなにかをしようと思うとき，そこには動機あるいは目的があります。この目的がなければなにも起こりません。因果律[1]はプログラムの世界でも健在なのです。自発的であれ強制的であれ，ある目的を達成するために私たちはプログラムを作ります。この目的達成のための手段がプログラミングであるといえます。

では目的達成のためには何が必要でしょうか。それは下準備です。たとえば，世界征服のためには悪の秘密結社を作らなければなりませんし，主砲を撃つためにはトランスフォーメーションしなければなりませんし，スペシウム光線を出すためには怪獣に痛めつけられなければなりません。着々と準備を重ねたあとで一気にケリをつける（実行する）のが目的達成のための一般的な過程です。

目的達成のための手段がプログラミングであるとすれば，プログラムとは目的達成のための過程を記述するものにほかなりません。したがってプログラムの中には「準備」と「実行」の段階が含まれていることになります。図1にプログラミングの一般的な概念を図示しておきます。この図がすべての基本です。

さて，プログラムの実行段階をもう少し詳しく眺めてみましょう。プログラムの実行はなんらかの効果をもたらします。つまり出力です。因果律の観点からは，そこになんらかの原因が必要です。つまり入力です。

出力としては，さまざまな演算結果，画面やファイルへのプリントなどが考えられます。一方，入力とはデータであり，そのデータはコンソール（キーボード）やファイルあるいは自分自身（プログラム内に格納されているか計算して作り出されるかのどちらか）から与えられます。そしてプログラムの実行とは，入力（原因）から出力（結果）を得るための操作であるともいえるのです。これを図2に示しておきます。

●変数

さて，これまでのことからプログラムとは，

1) 準備を行う部分と，

2) 与えられた入力を操作して出力を得る部分

から構成される目的達成のための手段ということができます。当然プログラムの主要部は入力から出力を得るための 2) の部分です。

ここでひとつの概念を学びましょう。それは変数です。変な数。いやそうではありません。内容が変わる数という意味ですね。内容とはデータのことで，中に入っているデータが変わる数が変数なのです。しかし，変数は数というよりも単にデータの「入れ物」と思ったほうがよいかもしれません。ある入力を操作してある出力を作るというプログラムの実行過程においては，入力となるデータの保存場所や出力されたデータの格納場所が必要となります。そのための役割を変数が果たすのです。

プログラム内で，

入力 → 操作 → 出力

という処理はひとつとは限りません。ある操作の出力が別の操作の入力になることもあります。プログラムの内部では「変数からデータを持ってきて別のデータに変換し別のあるいは同じ変数に格納する」という処理が至るところで行われているのです。

このように入力を出力に変換する小さな処理（ミクロな処理）がいくつも寄り集まって図2に示すようなプログラム全体の実行という処理（マクロな処理）が実現されるのです[2]。

したがって，それぞれの処理と処理の間でデータを受け渡しするための道具として変数が活躍するようになります。変数の役割を図3に示しておきましょう。

●副作用

ところで，ここまで読んできた人はある疑問を抱かれるかもしれません。たとえば，画面へのプリントを考えるとその場合は変数に格納すべきデータを出力してこないのです。この場合，プログラム全体のマクロな処理の実行結果として画面にプリントされる動作も出力には違いありません（図2は満たされる）が，プログラムを構成するミクロな各処理からの出力（データ出力）は存在しません（図3が満たされない）。

そこで図3を修正する必要があります。すなわち，プログラムの実行部の構成要素である「ミクロな処理には出力がないこともある」ということを付け加えます。ただし，この場合本当に出力が何もなければその処理は存在する意味がありません。存在しているからには何か理由があるのです。

実はこのような処理ではデータ出力がない代わりにプログラムの実行環境(コンピュータ)に対してなんらかの働きかけ（プリントせよ）を行っているのです。これは副作用[3]と呼ばれます。これはデータではなく，環境に対する指示の出力なのです。副作用を考慮して図3を書き換えたものが図4です。

---

1) 因果とは現在の不幸は前世の悪事の必然的結果であるということ。通常は悪い意味に使う。ここでは単に原因があるから結果があるという意味。

2) ドラクエⅣにおけるスライムとキングスライムの関係みたいなもの。じゃないかな。

3) 操作の結果を変数に代入するのではなく，一度に複数の変数の値を変える操作も副作用といえる。多くのプログラミング言語では，計算式による演算は通常はデータ出力のみで副作用を伴わない操作であるが，サブルーチンなどの副プログラムに関しては，データ出力を伴う操作を「関数」，副作用のみの操作を「手続き」といって2つを区別する。

プログラミング言語C

これまでの説明でプログラムというものの実体がわかってきたのではないでしょうか。つまり，変数（入力）を変数（出力）や副作用に変換する操作の集まりがプロ

図3　変数の役割（ミクロな処理）

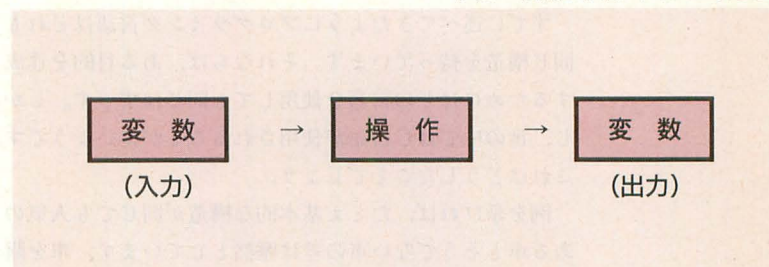
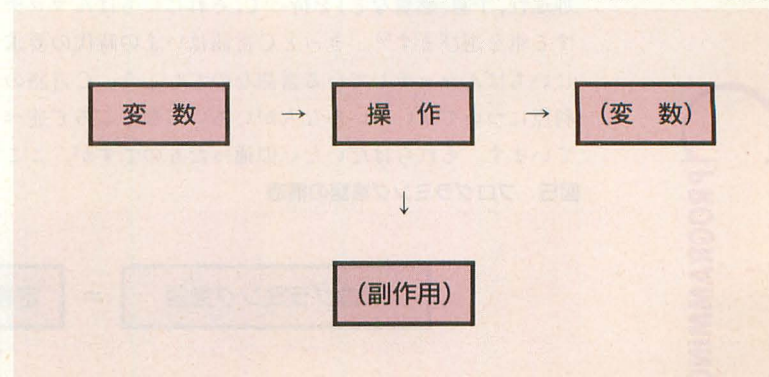



図4　プログラム実行のミクロな処理

グラムです。ここではそのようなプログラムを記述する言語という観点から見たときのC言語の特徴について考えてみましょう。

●プログラミング言語の構造

図1に示す操作（プログラミング）を実現するための言語がプログラミング言語です。そこには準備段階と実行段階があり，プログラミング言語もこの構造を反映したものでなければなりません。準備とは定義に対応します。変数や関数，手続き[4]の定義がこれに相当します。実行とは定義した変数，関数，手続きを使用した操作です。式の演算や関数呼び出しなどがこれに相当します。定義と操作，つまり準備と実行です。この2つを実現する手段がどんなプログラミング言語にも必ず備えられているのです。

しかし，実際のプログラミング言語には，さらにこの演算操作がどのような順序で実行されるべきか指定する機能が必要になります。条件分岐や繰り返しを指定する機能です。これらのことを考慮するとプログラミング言語は図5に示す構造を持っているといえるでしょう。したがって，あるプログラミング言語を学習するときは，

1）変数や関数の定義の方法
2）式の書き方や関数の呼び出し方
3）制御構造

がどうなっているかを知れば文法のほとんどを征服したことになります。

世の中に星の数ほどプログラミング言語はありますが，図5に示す構造を持たない言語はまずありません。ちょっと変わり者のLISPやPROLOGという言語でさえこの構造からはみ出すことはできないでいるのです。

●なぜC言語なのか

すでに述べてきたようにプログラミング言語はどれも同じ構造を持っています。それならば，ある目的を達成するためにはどの言語を使用しても同じはずです。しかし，世の中ではC言語が使用されることが多いようです。これはどうしたことでしょう。

例を挙げれば，たとえ基本的な構造が同じでも人気のある車とそうでない車の差は歴然としています。車を購入する人はたいていなんらかのポリシー（かっこよさ，加速性，予算，燃費など）を持って，それにいちばんマッチする車を選びます[5]。きっとC言語はいまの時代の要求にいちばんマッチしている言語なのでしょう。C言語の利点についてはいろいろな人がいろいろなところで述べています。それらはだいたい似通ったものですが，ここにおもいっきり主観を交えてまとめてみましょう。

1）覚えることが少ない

C言語はほかの言語と比べると制限がないのではと思えるほど自由な形式でプログラムを書くことができます。複雑な処理はすべて関数に任せて自分では何もしないという安易な（素晴らしい）性格の言語であるため，プログラムの記述に関して覚えることが少ないのがいいですね。

2）背景に崇高な思想がない

多くのプログラミング言語はコンピュータの素人に事務計算や数値計算を行わせるようにしたもので，変なことをしてコンピュータを暴走させてしまわないように厳しくエラーチェックをしています。またこれらの言語はプログラムのなんたるかを教えるためにお行儀のいい（厳しい）プログラミング作法を要求してくるのです。

しかし，C言語は生い立ちからして，OSを記術するために専門家が使うことを目的として作られたためエラーチェックをほとんどやってくれません。そこを逆手に取って技巧的なプログラミングも可能になるのです。プログラマが見逃してくれよと思うところはほとんど見逃してくれますし，プログラムを暴走させることも放題です。

3）便利なデータ構造を備えている

文法が単純な反面，データ構造は結構豊富です。特にポインタと構造体はほかの言語と比べると扱い方が単純なうえに強力なデータ処理を可能にしています。一見グータラで何もできないように思わせておいて実はキラリと光るものがある山岡士郎的言語なのです。

4）高級なアセンブリ言語として使用できる

本来C言語はミニコンのアセンブリ言語を高級言語のように使えるように拡張したものです。その文法にもアセンブリ言語の概念が少なからず反映されています。ところで，アセンブリ言語はCPUから見れば最強の言語です。それを高級言語のように使用できるのですからこんな心地いいことはありません。ここでいちばんの利点は開発期間の短縮です。それなりの性能を持ったプログラムを早期に開発できるのですから，多くの職業プログラマが飛びついたのも当然です。

5）アセンブリ言語への逃げ道を用意している

いくらアセンブリ言語に近いといっても，所詮はアセンブリ言語で（上手に）書いたプログラムにスピードもコードサイズも及びません。そこで，C言語ではアセンブリ言語のプログラムを容易に結合できるようになって

図5　プログラミング言語の構造

プログラミング言語 ＝ 定義 ＋ 操作 ＋ 順序制御

います。FORTRANなどでも裏技としてアセンブリ言語との混合を許す処理系がありますがこれは本道ではありません。しかし，C言語ではアセンブリ言語との混合は日常茶飯事に行われているので，それをやっても卑怯だと非難されることはありません。

●C言語は初心者に有用か

これまでに示した特徴を見るとC言語はかなりいい加減な言語であるといえます。記述形式が自由だ（何をやってもいい）という点は「型」から入ってくる初心者プログラマにとっては戸惑いの種でしかありませんし[6]，すぐさま暴走したのではデバッグが大変です。また，C言語はアセンブリ言語を拡張したものですから，高級言語一本槍でアセンブリ言語の経験のない人には理解できない概念がいくつかあります。

4～5年前の某雑誌に載ったC言語の解説記事を読み返してみると「C言語はlint[7]なしでは使えたものではない」ということが至るところで述べられています。これはあまりにいい加減な(これまでの高級言語とは違う)言語がもてはやされていることに対する非難だったのではないでしょうか。個人的にはこの意見はC言語の本質を見ていない的外れなものだと思いますが，それまではかの言語に習熟していた人たちの目にそう映ったのも無理もないことと思います。

しかし，そういう専門家の危惧とは裏腹にそのいい加減である点を理由として，すなわち，いい加減であるがゆえになんでもありなのを理由として，C言語が流行してきたことは覚えておく必要があるでしょう。

確かにC言語の流行を支えた人は初心者ではありません。C言語をアセンブラの代用品[8]として使用する人がその便利さを実感することで自然と流布していったことに間違いはありません。それならば，初心者はC言語を使ってはいけないのでしょうか。答えはもちろんノーです。

C言語は既成の型にとらわれない自由な考え方でプログラミングするのに役立ちます。いくら暴走させたっていいじゃありませんか。きっとその経験が後で役立つときも来るでしょう。あるいは，C言語には文部省特選(?)の構造化プログラミングを行うための仕組みも用意されていますから，FORTRANやPASCALのようなお行儀のいい高級言語としてC言語を学び始めることも可能です。この場合，ただちにC言語の持つ恩恵をすべて得られるわけではありませんが，徐々にC言語の本質に迫っていけばいいでしょう。それでも，ほかの言語を学ぶよりも＋αの利益があるに違いありません。

結局，いまさらFORTRANやPASCALなどを学ぶくらいならC言語を高級言語のひとつとして学んだほうがいいということですね。それで，この連載では当面は初心者のための高級言語としてのC言語入門を目指していきたいと思います。

---

4) やばい，関数や手続きがどういうものか概念を説明してなかった。関数や手続きとはプログラム内に何度も共通に現れる操作に名前をつけて別の場所で定義し，その名前で参照できるようにしたもの。構造化プログラミングの観点からは特に共通な操作でなくてもモジュール化のために関数や手続きが定義される。詳しくはおいおい説明する予定。

5) 最近ではシルビアとレビンが街中（東京近辺）に溢れてるような気がする。

6) ドラクエIIIで順々に課せられる小さなクエストをクリアしながら船を手に入れるまではよかったが，それ以降はこれといった目的が与えられてないため戸惑ってしまう人が多いのと同じ。

ところで，ソフトウェア会社では字下げの文字数，改行の仕方，名前のつけ方などの形式が厳密に規定されているので自由にプログラムを書くことはできない。こんなのはいやだ。

7) lintはコンパイラのチェックしない，引数の個数や型の一致，値が代入されないままで使用されている変数など，文法エラーではないが意味的におかしな部分（もちろん文法エラーも）を教えてくれるツール。最近はlint以上に口うるさいコンパイラが多くなった。

8) 本誌1989年12月号の特集記事（p.31）での祝一平氏による，「地獄のような86系CPUのアセンブリ言語の代用品としてC言語がクローズアップされた」という説は結構本当かもしれない。

C言語のプログラム

今回はイントロダクションということで，C言語の解説を本格的に行うことはしません。それは来月から徐々にやっていきましょう。ここではC言語のプログラムが図5に示す構造を持っていることを確かめておきます。

●C言語のプログラムの構造

図5に示すようにプログラム言語は「定義」，「操作」，「順序制御」の3つの部分から成り立っています。これをC言語の文法でいえば，次のように言い換えることができます。

定義→変数の定義，関数の定義
操作→式，関数呼び出し
順序制御→文

まだ具体例がないので唐突かもしれませんが，これがC言語の本質なので覚えておけばのちのち役に立つと思います。また，関数が何度も顔を出してきますが，C言語では関数が重要な役割を果たしている[9]ことにもチェックを入れておきましょう。ここでいう関数にはライブラリ関数も含まれます。C言語では難しそうな処理はすべてライブラリ関数で行うという傾向があります。ライブラリ関数とはあらかじめ定義されている（ユーザーは定義する必要のない）便利でよく使われる関数のことで，プログラム中で自由に呼び出して使うことができます。

このように，「定義」，「操作」，「順序制御」の3つの内容（具体的には関数と変数と文と式）さえ理解すればC言語の文法はおしまいです。こう考えてくると，一見は難しそうなC言語でも簡単に覚えられるような気がしませんか。

●実例に見るプログラムの構造

それでは，実際のC言語のプログラムによって，これまでに述べてきたプログラムの構造を持っていることを

見ていきましょう。リスト1がC言語の典型的プログラムです。プログラムの各部分が何を行っているかはコメントを付けておいたのでわかるでしょう。とはいえ，C言語を知らない人にはまったくハナモゲラな計算式やプログラム上の表現があるかもしれませんね。要するに，プログラムでは変数（a,b,c,d）と関数（main,max2）が定義され，変数に入れられた値が定義された関数やライブラリ関数（scanf,printf）や演算によって操作されたり副作用を生み出されていることを感じることができればいいと思います。これが形式的には図5に示す構造を持ち，内容的には図2から図4に示す概念を持っていることがわかると思います。

●プログラムの実行はmainを実行すること

さて，リスト1のプログラムは変数と関数の定義をして，さらにそれらを実行するような関数を定義したものです。一応，図2から図5に示す形式をしていますが，一歩下がって見るとプログラム自身は変数と関数の定義にすぎません。それでは定義された関数（リスト1のプログラム全体）を実行するとはどういうことなのでしょう。実はそれこそがプログラムを実行するということで，Cコンパイラでコンパイルしてできる実行形式のプログラムを実行するということなのです。

C言語のプログラムにはmainという関数が必ずひとつだけあります。プログラムの実行とはこのmainという関数を実行することなのです。したがって，プログラムの実行時の動作を追うためにはプログラムの中でmainという関数が何をしているかを調べます。

リスト1において，mainという関数ではscanf関数を呼んで，乗算の計算をやって，max2関数を呼んで，最後にprintf関数を呼んでいますね。プログラムの実行ではこれが順に実行されるのです。

リスト1のプログラムをコンパイル[10]して実行するとscanf関数によって入力待ちになります（プロンプトは出ない）から適当な2つの整数をスペースで区切って入力してください。式とmax2関数で計算された，2つの整数の積と最大値がprintf関数によって画面にプリントされます。

9）かつて某氏がC言語を関数型言語であると本誌の特集記事で紹介したことがあった。これに対して別の某氏はC言語は手続き型言語だと憤慨していた。C言語は関数型言語であるLISPに非常に近いので，あながち間違いではないと私は思うのだが。

10）コンパイルの仕方はいまさら説明しなくてもいいかな。Cコンパイラを持っていてコンパイルを一度もしたことのない人がいるのだろうか。これくらいはマニュアルを読みましょう。

＊

今回はC言語の連載の予告編という意味合いを込めて，私が日頃感じていることを書き綴ってみました。プログラミング言語なんてどれも同じようなもので基本をマスターすればあとはみな同じ，でもC言語が少しだけ有利かなというのが結論です。今回は観念的に走り過ぎたきらいがありますが，来月からは明るくて楽しいまっとうなC言語の連載にしたいと思っています。

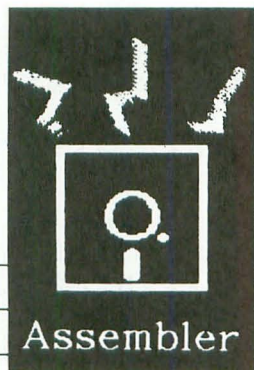
ところで，みなさんがC言語に対して抱いている疑問があればどしどし編集部まで送ってください。珍問，難問，奇問なんでも結構です。特にみなさんがつまずきやすい問題に関してはできる限り答えていきたいと思っています。

リスト1

```
 1: /*
 2:          この例はプログラムの基本的な構造を示している
 3:                  (これはコメントです)
 4: */
 5:
 6: int a, b, c, d;                  /*  変数の宣言（定義） */
 7:
 8: main()                           /*  関数の定義  始まり */
 9: {
10:    scanf("%d %d", &a, &b );  /*  関数呼び出し  ライブラリ関数      */
11:                              /*  戻り値：なし（本当はあるけど無視）*/
12:                              /*  副作用：変数 a，b に値を入れる    */
13:
14:    c = a * b;                /*  変数を演算して変数に代入する      */
15:
16:    d = max2(a, b);           /*  関数呼び出し  ユーザー定義関数    */
17:                              /*  戻り値：引数 a，b の最大値        */
18:                              /*  副作用：なし                      */
19:
20:    printf("a=%d b=%d¥n", a, b);/*  関数呼び出し  ライブラリ関数   */
21:    printf("a*b=%d¥n", c);      /*  戻り値：なし                   */
22:    printf("max2(a,b)=%d¥n", d);/*  副作用：変数の値をプリントする */
23:
24: }                                /*  関数の定義  終わり     */
25:                                  /*  戻り値：なし           */
26:
27:
28: max2(x, y)                       /* 関数の定義  始まり                */
29: int x, y;                        /* 引数の定義（そのうち説明します）  */
30: {
31:    return(((x+y)+abs(x-y))/2);   /* 変数と別の関数呼び出し（abs）の値 */
32:                                  /* との演算を関数の戻り値にして返す  */
33: }                                /* 関数の定義  終わり                */
34:                                  /* 戻り値：return 文で指定した値     */
```

画像に変化を与える処理

Murata Toshiyuki 村田 敏幸

今月はちょっとした画像処理に挑戦してみましょう。色の性質とデータの関係を知れば，お馴染みの論理演算やビット操作によって，さまざまな画像処理を行うサブルーチンが作れます。応用範囲も多彩ですからぜひともものにしてください。

今回は先月の矩形領域塗り潰しの応用として，グラフィック画面の任意の矩形領域に含まれる各点の色をある規則によって置き換えるサブルーチンをいくつか作ってみる。色の反転（補色に置き換える），色の量子化（階調を落とす），モノクロ化，色の平均化（ボカし），輪郭抽出といったあたりだ。

色とコード

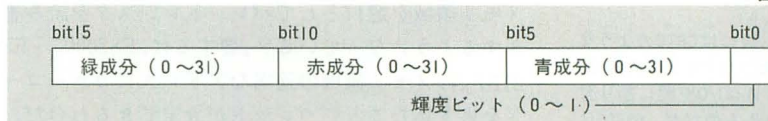
最初にX68000上での色の扱いについておさらいしておこう。カラーコードとパレットコードの意味は明確にしておきたい。

カラーコードは色そのものにつけられた番号であり，色とカラーコードは1対1に対応する。X68000のカラーコードはRGB[1]ごとの輝度各5ビット＋RGB共通の輝度1ビットからなる計16ビットで表現される（図1）。R，G，Bそれぞれにつき，その色成分をどのくらい含むかが0～31の32段階で表され，組み合わせで32×32×32＝32768色，これに輝度ビットが0か1かが加わって65536色となる[2]。

ここでリスト1のGMACRO.Hを見てもらおう。今月のプログラムで使うために用意したマクロ定義ファイルだ。この下のほう，57～75行にカラーコードをRGBに分解するマクロDERGBと，RGBの色成分からカラーコードを得るマクロRGBが定義されている[3]。グラフィック処理ではよく出てくる操作なので，図1と見比べて動作をしっかりつかんでおいてほしい。ビットシフト，ビットマスク，ビット列のはめ込みといったビットレベルの操作のオン

1) RGBはいわずと知れた赤・緑・青の光の3原色だ。
2) カラーコードの輝度ビットを立てると，RGBそれぞれが0.5だけ明るくなる。
3) RGBからカラーコードへの変換はIOCSコールでもサポートされてはいるのだが，マクロのほうがパラメータの渡し方の制限が緩いぶん，利用しやすい。

図1

bit15	bit10	bit5	bit0
緑成分（0～31）	赤成分（0～31）	青成分（0～31）	輝度ビット（0～1）

リスト1 GMACRO.H

```
 1: IMASK    equ     %00000_00000_00000_1
 2: BMASK    equ     %00000_00000_11111_0
 3: RMASK    equ     %00000_11111_00000_0
 4: GMASK    equ     %11111_00000_00000_0
 5: *
 6: RGBMAX   equ     31
 7: RGBGRAD  equ     RGBMAX+1
 8: *
 9: ABS      macro   X                   *abs(x)
10:          local   done
11:          tst.w   X
12:          bpl     done
13:          neg.w   X
14: done:
15:          endm
16: *
17: SGN      macro   X                   *sgn(x)
18:          local   min,done
19:          tst.w   X
20:          bmi     min
21:          beq     done
22:          move.w  #1,X
23:          bra     done
24: min:     move.w  #-1,X
25: done:
26:          endm
27: *
28: MAX      macro   Y,X                 *max(x,y)
29:          local   done
30:          cmp.w   Y,X
31:          bge     done
32:          move.w  Y,X
33: done:
34:          endm
35: *
36: MIN      macro   Y,X                 *min(x,y)
37:          local   done
38:          cmp.w   Y,X
39:          ble     done
40:          move.w  Y,X
41: done:
42:          endm
43: *
44: MINMAX   macro   Y,X                 *x=min(x,y),y=max(x,y)
45:          local   skip
46:          cmp.w   Y,X
47:          bge     skip
48:          exg.l   Y,X
49: skip:
50:          endm
51: *
52: MEAN     macro   Y,X                 *x=(x+y)/2
53:          add.w   Y,X
54:          lsr.w   #1,X
55:          endm
56: *
57: DERGB    macro   COL,B,R,G
58:          lsr.w   #1,COL              *COL = 0GGGGGRRRRRBBBBB
59:          move.w  COL,B               *  B = 0GGGGGRRRRRBBBBB
60:          lsr.w   #5,COL              *COL = 000000GGGGGRRRRR
61:          move.w  COL,R               *  R = 000000GGGGGRRRRR
62:          lsr.w   #5,COL              *COL = 00000000000GGGGG
63:          move.w  COL,G               *  G = 00000000000GGGGG
64:          andi.w  #RGBMAX,R           *  R = 00000000000RRRRR
65:          andi.w  #RGBMAX,B           *  B = 00000000000BBBBB
66:          endm
67: *
68: RGB      macro   B,R,G,COL
69:          move.w  G,COL               *COL = 00000000000GGGGG
70:          lsl.w   #5,COL              *COL = 000000GGGGG00000
71:          or.w    R,COL               *COL = 000000GGGGGRRRRR
72:          lsl.w   #5,COL              *COL = 0GGGGGRRRRR00000
73:          or.w    B,COL               *COL = 0GGGGGRRRRRBBBBB
74:          add.w   COL,COL             *COL = GGGGGRRRRRBBBBB0
75:          endm
76: *
77: RGBtoY   macro   B,R,G,Y
78:          add.w   G,G                 *G = 2G
79:          add.w   R,G                 *G = 2G+R
80:          move.w  G,Y                 *Y = 2G+R
81:          add.w   Y,Y                 *Y = 2(2G+R)
82:          add.w   G,Y                 *Y = 3(2G+R)
83:          add.w   B,Y                 *Y = 6G+3R+B
84:          ext.l   Y
85:          divu.w  #10,Y               *Y = (6G+3R+B)/10
86:          endm
```

パレードだから，このあたりに不安のある人は注釈を参考にデータの変化を追ってみること。

色につけられた絶対的な番号であるカラーコードに対して，パレットコードのほうはいってみるなら画材のパレット（一枚板のじゃなくて小学校の図工の時間に使ったような枠がたくさんあるヤツ）の枠についた番号であり，特定のパレットコードがどの色にあたるかは決まっていない。パレットの設定(枠に置いた絵の具の色)によってどうにでも変わる。ただし，画面モードをIOCSなどによって初期化したときにはあたかもパレットコードの各ビットが，

GGGGGRRRRRBBBBBI (65536色モード時)
GGRRRBBB (256色モード時)
GRBI (16色モード時)

のような意味を持つかのように初期化される。とくに65536色モードにおける初期状態ではパレットコード＝カラーコードとなる。

G-RAMに書き込むデータはパレットコードであり，ディスプレイへの表示の際にはCRTC（CRT Controler：画面制御を行うLSI）がG-RAMの内容を読み出したうえでパレットの設定を参照し，カラーコードに変換してくれている。こういう仕組みだから，パレットを変更すると瞬時に画面の色も変わるというわけだ。

パレットの設定はCRTCの内部レジスタを書き換えることで行う。CRTCのパレットレジスタはスーパーバイザ空間のE82000H番地以降256ワードにマッピング(割り付け)されており，68000からはこのメモリ領域を窓口としてパレットレジスタを読み書きするようになっている[4]。要するに，E82000H～E821FFHのメモリ領域の適当なアドレスにカラーコードを書き込むことでパレットが設定できるわけだ。

16色モードや256色モードではE82000H番地からの1ワードがパレットコード0，E82002H番地からがパレットコード1……というようにアドレスとパレットコードは素直に対応している。

リスト2　GSAVEPAL.S

```
 1:          .include        doscall.mac
 2:          .include        gconst.h
 3: *
 4:          .xdef   gsavepalet
 5:          .xdef   gloadpalet
 6: *
 7:          .text
 8:          .even
 9: *
10: gsavepalet:
11: FNO = 8
12:          link    a6,#0
13:
14:          move.l  #256*2,-(sp)
15:          pea.l   GPALET
16:          move.w  FNO(a6),-(sp)
17:          DOS     _WRITE
18:
19:          unlk    a6
20:          rts
21: *
22: gloadpalet:
23: FNO = 8
24:          link    a6,#0
25:
26:          move.l  #256*2,-(sp)
27:          pea.l   GPALET
28:          move.w  FNO(a6),-(sp)
29:          DOS     _READ
30:
31:          unlk    a6
32:          rts
33:
34:          .end
```

アドレス＝E82000H＋パレットコード×2

だ。ところが，65536色モードではパレットレジスタの数が全然足りないので，かなり変則的な形でパレットが実現されている。65536色モードでパレットを変更したいという場面はあまりないとは思うが，簡単にまとめておこう。

4桁の16進数mmnnHで表されるパレットコードには，

H=E82002H＋(mmH&FEH)×2＋(mmH&1)
L=E82000H＋(nnH&FEH)×2＋(nnH&1)

の2つのアドレスが対応し，アドレスHで指定される1バイトにカラーコードの上位バイト，アドレスLにカラーコードの下位バイトが格納される。明らかにパレットコードmmXXHで表される256色についてアドレスHは共通となり，パレットコードXXnnHとアドレスLについても同様の256対1の対応が見られる。このため，パレットコードmmnnHのパレット設定を変更すると，パレットコードmmXXHで表される256色とXXnnHで表される256色の計511色（1色はダブり）の設定も同時に変わってしまうことになる。この様子は，

```
int x,y
screen 1,3,1,1
for y=0 to 255
for x=0 to 255
    pset ( x,y,(y shl 8) +x )
next
next
```

のようなX-BASICプログラムで画面に65536色を表示したうえで適当にパレットを変更してみるとわかるだろう。

本筋からは離れるのだが，パレットの話の最後にパレットのセーブ・ロードを行うサブルーチンを示しておく（リスト2）。入出力先のファイルハンドルをスタックに積んで呼び出す。見てのとおり，DOSコールWRITEとREADでE82000H番地以降の512バイトをセーブ・ロードしているだけだ。読み書きの過程ではスーパーバイザ空間をアクセスすることになるわけだが，このサブルーチンはユーザーモードで呼び出してもちゃんと動く(問1：なぜか？)。

では，ぼちぼちと本題に入ろう。が，その前にひとつだけ断っておかなければなるまい。これから作るプログラムは基本的には65536色モードにしか対応していない。また，パレットは初期状態のままであることを前提としている。G-RAMに書き込まれているパレットコード＝カラーコードを仮定しないと，処理がややこしく（ものによっては不可能に）なってしまうからだ。

色の反転と量子化

まず，色の反転と量子化をまとめて片づけてしまおう。実はどちらも先月作ったプログラムを利用することで実現できる。

4) 一般にはCRTCのような(プロセッサから見ての)外部デバイスはI/O空間に割り付けられるものだが，68000はモトローラの伝統でI/O空間というものを持たず，I/Oをメモリにマッピングする“メモリマップドI/O（memory mapped I/O)”が採用されている（英語の発音に従うならメモリマップトと表記すべきだが，ここでは“日本古来”の伝統によりメモリマップドと濁った表記を採用した)。

正しい表現ではないのかもしれないが，ここでは色の反転＝補色への置き換えと考える。補色とは補い合うことで無彩色となるような色で，加法混色[5]における補色と減法混色[6]における補色があるが一般には前者のことを指す。白から引いた残りの色といえばわかるだろう。カラーコードのレベルでは，補色を求める操作はカラーコードの各ビットを反転する処理に置き換えて考えることができる（問2：なぜか?）。

任意の16ビットデータは$FFFF_H$とXORをとるとビットが反転するから，先月作ったXORモードのボックスフィルルーチンで描画色として$FFFF_H$を指定すれば色の反転が行えることになる。パラメータのつじつまを合わせてgfill_xorを呼び出すだけのものだが，一応リスト3に矩形領域の色反転を行うサブルーチンを示しておく。メインルーチンは用意していないから，前回や前々回のプログラムを適当に流用してもらいたい。

続いては色の量子化，RGB各32階調から16階調や8階調なんかに階調を落とす操作を考える。絵の質をなるべく損なわないようにして色数を減らすのではなく，単純に微妙な色の変化を切り捨てる。当然，階調を落とすほど絵は粗くなっていく。場合によっては画像圧縮の前処理に使えないこともないが，特殊効果だと思ってもらったほうがいい。色の量子化はカラーコードのレベルで考えると，RGBそれぞれの下位ビットをマスクする処理に相当する。ビットマスクといえばAND，ということは前回のANDモードのボックスフィルルーチンが利用できる。RGBに分解してから下位ビットをマスクするまでもなく，カラーコードと，

1111011110111100_B

でANDをとれば各色16階調，以下，

1110011100111000_B　　8階調

1100011000110000_B　　4階調

1000010000100000_B　　2階調

となる。プログラム例は示す必要もないだろう。

なお，ANDモードのボックスフィルルーチンはほかにも，描画色に，

1111111111111110_B

を指定することで輝度ビットだけをマスクしたり(65536色から32768色への変換)，

0000000000111110_B

0000011111000000_B

1111100000000000_B

によってそれぞれ青，赤，緑成分だけを残したりといった用途にも使える。

モノクロ32階調に

今度は画像のモノクロ化だ。モノクロといっても白黒の2値化ではなく，グレイスケール（黒～灰色～白に段階的に変化するような色の並び）に変換することを考える。X68000では単色であれば輝度ビットを使って64階調を出すこともできるが，ここでは輝度ビットを無視した32色のみを使うことにしよう。アルゴリズムとしては，各点ごとに明るさ（仮にLとおく）を32階調で求め，R=G=B=Lとなるような色に置き換えていけばよい。肝心の明るさを求める部分だが，とりあえずはRGBのうち最大のものを使うこととしよう。プログラムはリスト4のようになった。

下準備部分やメインループの組み方などの基本部分は前回のリストとほぼ同じだ。パラメータを取得し実画面の範囲でクリッピングしてから領域の左上隅のG-RAMアドレスを求め，y座標とx座標に関する2重ループの中で1ドットずつ処理していく。ただ，外側のy座標に関するループはほんのわずかに最適化されている。見比べてみて，違いとその意

5) 光は混ぜ合わせることによって明るくなり，白に近づいていく。だから加法混色。
6) 絵の具は混ぜ合わせることによって（吸収される光が多くなるため）暗くなり，黒に近づいていく。だから減法混色。減法混色系の3原色シアン・マゼンタ・黄色（青・赤・黄は不正確）は加法混色系の3原色である赤・青・緑の補色となっている。

リスト3　GNEGATE.S

```
 1:          .include        gconst.h
 2: *
 3:          .xdef   gnegate
 4:          .xref   gfill_xor
 5: *
 6:          .offset 0
 7: *
 8: X0:      .ds.w   1
 9: Y0:      .ds.w   1
10: X1:      .ds.w   1
11: Y1:      .ds.w   1
12: *
13:          .text
14:          .even
15: *
16: gnegate:
17: PARPTR = 12
18:          movem.l a1,-(sp)
19:          link    a6,#0
20:
21:          move.l  PARPTR(a6),a1
22:          move.w  #$fffe,-(sp)
23:          move.l  4(a1),-(sp)
24:          move.l  (a1),-(sp)
25:          move.l  sp,-(sp)
26:          bsr     gfill_xor
27:
28:          unlk    a6
29:          movem.l (sp)+,a1
30:          rts
31:
32:          .end
```

リスト4　GMONO.S

```
 1:          .include        gconst.h
 2:          .include        gmacro.h
 3: *
 4:          .xdef   gmonotone
 5:          .xref   gramadr
 6:          .xref   gfclip
 7:          .xref   grayscale
 8: *
 9:          .offset 0
10: *
11: X0:      .ds.w   1
12: Y0:      .ds.w   1
13: X1:      .ds.w   1
14: Y1:      .ds.w   1
15: *
16:          .text
17:          .even
18: *
19: gmonotone:
20: PARPTR = 8
21:          link    a6,#0
22:          movem.l d0-d7/a0-a1,-(sp)
23:
24:          move.l  PARPTR(a6),a1    *a1=パラメータ受け渡し領域
25:          movem.w (a1),d0-d3       *d0-d3に座標を取り出す
26:
27:          bsr     gfclip           *クリッピングする
28:          bne     done             *Z=0なら描画の必要なし
29:
30:          bsr     gramadr          *G-RAM上のアドレスを得る
31:
32:          sub.w   d0,d2            *d2=横ドット数-1
33:          sub.w   d1,d3            *d3=縦ドット数-1
34:
35:          move.w  #GNBYTE-2,d1     *d1=ライン間のアドレスの差
36:          sub.w   d2,d1            *(右端から下のラインの
37:          sub.w   d2,d1            *左端まで)
38:
39:          lea.l   grayscale,a1
40: loop1:   move.w  d2,d4            *d4=横ドット数-1
41: loop2:   move.w  (a0),d0          *カラーコードを取り出し
42:          DERGB   d0,d5,d6,d7      *RGBに分解する
43:          MAX     d6,d5            *R,G,Bの最大値を
44:          MAX     d7,d5            *  d5に求め
45:          add.w   d5,d5            *rgb(d5,d5,d5)の色で
46:          move.w  0(a1,d5),(a0)+   *  点を打つ
47:          dbra    d4,loop2         *繰り返す
48:          adda.w  d1,a0            *すぐ下のラインへ
49:          dbra    d3,loop1         *繰り返す
50:
51: done:    movem.l (sp)+,d0-d7/a0-a1
52:          unlk    a6
53:          rts
54:
55:          .end
```

7) ちなみにこのテーブルはX-BASICで短いプログラムを書いて計算させ，ファイルに出力したものだ。32個程度のテーブルであれば手作業で作成できる範囲だが，プログラムにやらせたほうが間違いがない。

味を把握してもらいたい。

41行以下が1ドット分の色変換を行う部分だ。41～42行で色コードを取り出しRGBに分解してから，マクロMAXでまず青成分と赤成分の大きいほうをd5に求め（43行），続いて緑成分とd5の大きいほうをd5に残す（44行）。結果的にRGBのうち最大のものがd5に得られた。これからR=G=B=d5となる色を求めるわけだが，リスト4ではその手間を惜しんで，あらかじめ作成しておいた32色の色テーブル[7]から適切な値をひいてくるという手を使っている（45～46行）。テーブル自体はほかにも使い道がありそうだから別ファイル（リスト5）にしておいた。

さて，リスト4ではとくに根拠もなくRGBのうち最大のものをその点の明るさだとみなした。妥当な線だとも思えるが，実際にはなにかと不都合がある。もっとも明るい赤と青で塗り分けられた部分は真っ白に変換されてしまうし，また，変換の前後で明るさのバランスが崩れ，ときには明暗が逆になったように見えることもある。切り捨てた情報量も大きいが，これは人間の目（そして脳）が純粋な光の強弱だけで明るさを判断しているのではないため，ということらしい。画像をモノクロに変換するときに，この人間の視覚系の性質を考慮すればより自然な変換が行えるだろう。これには，カラーテレビ放送の仕組みが参考になる。詳しくは稿末に挙げた参考文献を見てもらうことにし，ここではプログラムを書くのに必要なだけの情報を示しておく。

アメリカや日本などのテレビ放送では画像をYIQの3信号に分解して送る方式を採用している。大雑把にいうとYが輝度，Iが人間の目にとって重要な（色の違いに敏感な）肌色を含むオレンジからシアンにかけての色調，Qはそれ以外の色を表す。テレビの受像機側ではこれをRGBに再構成して画面に映している。3つの信号に分けるのなら最初からRGBに分解しておけばいいように思えるが，そうなっていないのには白黒テレビとの互換性を保つ必要があったこと（白黒放送で使われていた電波の帯域にY信号を乗せれば白黒テレビでもカラー放送の電波を受像できる）と，放送に使える電波の帯域に制限がある（人間の目にとって重要でない色は狭い帯域に押し込むことで帯域を節約できる）という2つの理由による。

ここで，Yはちゃんと人間の主観的な明るさの感覚を合わせて調整してあるというのがポイントだ。ということは，画像の各点の色をRGBからYIQに変換し，Yの値に応じた灰色に置き換えるようにすれば，それらしいモノクロ画像が得られるだろう。件の文献を見れば，

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & 0.522 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

という式まで載っている。いま必要なのはYだけだから，

$$Y=0.299R+0.587G+0.114B$$

となる。

この式をマシン語で計算するにあたっては小数が入っているのがやっかいだ。しかし，どうせYは0～31の整数で求めるのであり，精度を追求することにあまり意味はない。そこで係数を適当にまるめたうえで両辺を10倍し，

$$10Y=3R+6G+B$$

として整数演算に帰着させる。右辺を計算した結果はYを10倍したものだから，最後にこの値を10で割ればYが得られる。

リスト6に画像モノクロ化サブルーチンの新しい版を示す（リスト4からの変更部分のみ）。リスト4の版にも使い道はありそうだから，サブルーチン名は変えておいた。Yの値を求めるにはリスト1で定義されたマクロRGBtoYを利用しているので，ソース上での修正点はごく僅かだ。マクロRGBtoYの中では例によって乗算を加算に置き換えて計算することで遅い乗算命令を使わずにすまし，さらには式を，

$$10Y=3(R+2G)+B$$

と変形して計算することで加算の回数も減らしている。

リスト5 GRAYSCALE.S

```
 1:          .xdef    grayscale
 2: *
 3:          .data
 4:          .even
 5: *
 6: grayscale:
 7:          .dc.w    $0000,$0842,$1084,$18c6
 8:          .dc.w    $2108,$294a,$318c,$39ce
 9:          .dc.w    $4210,$4a52,$5294,$5ad6
10:          .dc.w    $6318,$6b5a,$739c,$7bde
11:          .dc.w    $8420,$8c62,$94a4,$9ce6
12:          .dc.w    $a528,$ad6a,$b5ac,$bdee
13:          .dc.w    $c630,$ce72,$d6b4,$def6
14:          .dc.w    $e738,$ef7a,$f7bc,$fffe
15:
16:          .end
```

リスト6 GMONO Y.S

```
 4:          .xdef    gmonotone_y
19: gmonotone_y:
41: loop2:   move.w   (a0),d0            *カラーコードを取り出し
42:          DERGB    d0,d5,d6,d7        *RGBに分解する
43:          RGBtoY   d5,d6,d7,d0        *Yをd0に求め
44:          add.w    d0,d0              *rgb(d0,d0,d0)の色で
45:          move.w   0(a1,d0),(a0)+     *  点を打つ
46:          dbra     d4,loop2           *繰り返す
47:          adda.w   d1,a0              *すぐ下のラインへ
48:          dbra     d3,loop1           *繰り返す
49:
50: done:    movem.l  (sp)+,d0-d7/a0-a1
51:          unlk     a6
52:          rts
53:
54:          .end
```

ソフトフォーカス?

次はボカしだ。概念的には色を周囲に少しずつにじませてやればよいわけだが，プログラムにするうえでは逆に周囲の点から少しずつ色をもらってくると考えたほうがわかりやすい。ボカし具合は色を混ぜ合わせる比率と色をにじませる範囲で調節できる。広い範囲ににじませる場合には，中心から遠い点からは色を少しだけもらい，近くの点からはたくさんもらうというように適当な重みをつけて平均化してやると綺麗にボケる。このとき，にじませる範

囲は正方形にとるよりは円形にしたほうがより自然な結果が得られる。

という話をしておきながら，リスト7に示すサブルーチンはかなりいいかげんな簡略版になってしまった。色をにじませる範囲は上下左右の1ドット，混合の比率は中心4：周囲1固定となっているし，ほかにもぽつぽつと手抜きがある。ひとつの例だと思ってもらえればありがたい。気乗りはしないが，一応簡単にプログラムの説明をしておく。

先に114行以下のサブルーチンを見てもらおう。ここでは中心の色と周囲の色のRGBごとの混ぜ合わせを行っている。ボカし処理の本体だ。サブルーチンが呼び出された時点で，いま注目している点とその周囲の点の色が，a3レジスタでポイントされるメモリ領域に18〜25行のような順序・構造で格納されている。中心の点の色をRGBに分解してから4倍し，これに周囲の色をRGBごとに足し込んで，8で割れば色を4:1:1:1:1で混ぜ合わせたことになる。

62行からメインループが始まる。色の変換自体はサブルーチンにまかせているので，メインループでは処理対象となる5ドットの色をバッファにセットしてサブルーチンに渡す以上の仕事はしていない。例によって，処理は左上の点から右方向，そして下方向に進む。ここで，処理途中におけるある1点に注目しよう。すぐ左の点と真上の点はすでにボカし処理済みであり，この点の色を拾ってしまうと正しい色の平均化が行えないことがわかる。そこで，リスト7では真上の1ライン分は142行で確保したバッファに保存しておき，左の点に関しては画面の書き換えを1拍遅らせる（72〜73行）ことで対処している。

また，リスト7では，指定された領域の一番上と一番下の1ライン，および各ラインの左端と右端を特別扱いし，本来であれば指定された矩形領域の1ドット外側の点の色を拾うべきところを1ドット内側の点の色で代用していたりする。これは指定の矩

リスト7　GSOFT.S

```
  1:         .include        gconst.h
  2:         .include        gmacro.h
  3: *
  4:         .xdef   gsoftfocus
  5:         .xref   gramadr
  6:         .xref   gfclip
  7: *
  8: DR      equ     2       *右の点とのアドレスの差
  9: DD      equ     GNBYTE  *下の点とのアドレスの差
 10: *
 11:         .offset 0
 12: *
 13: X0:     .ds.w   1
 14: Y0:     .ds.w   1
 15: X1:     .ds.w   1
 16: Y1:     .ds.w   1
 17: *
 18:         .offset 0
 19: *
 20: C:      .ds.w   1       *中心
 21: L:      .ds.w   1       *左
 22: U:      .ds.w   1       *上
 23: R:      .ds.w   1       *右
 24: D:      .ds.w   1       *下
 25: PBUFSIZ:
 26: *
 27:         .text
 28:         .even
 29: *
 30: gsoftfocus:
 31: PARPTR = 8
 32:         link    a6,#-PBUFSIZ
 33:         movem.l d0-d7/a0-a2,-(sp)
 34:
 35:         move.l  PARPTR(a6),a1   *a1=パラメータ受け渡し領域
 36:         movem.w (a1),d0-d3      *d0-d3に座標を取り出す
 37:
 38:         bsr     gfclip          *クリッピングする
 39:         bne     done            *Z=0なら描画の必要なし
 40:
 41:         bsr     gramadr         *G-RAM上のアドレスを得る
 42:
 43:         sub.w   d0,d2           *d2=横ドット数-1
 44:         sub.w   d1,d3           *d3=縦ドット数-1
 45:
 46:         move.w  #GNBYTE-2,d1    *d1=ライン間のアドレスの差
 47:         sub.w   d2,d1           *(右端から下のラインの
 48:         sub.w   d2,d1           *左端まで)
 49:
 50:         movea.l a0,a1           *一番上のラインを
 51:         lea.l   lbuff,a2        *  バッファにコピーしておく
 52:         move.w  d2,d4           *
 53: loop0:  move.w  (a1)+,(a2)+     *
 54:         dbra    d4,loop0        *
 55:
 56:         subq.w  #1,d2           *d2=横ドット数-1-1
 57:         bmi     done
 58:         subq.w  #1,d3           *d3=縦ドット数-1-1
 59:         bmi     done
 60:
 61:         lea.l   -PBUFSIZ(a6),a1
 62: loop1:  move.w  d2,d4           *d4=横ドット数-1-1
 63:         lea.l   lbuff,a2        *a2=1ラインのバッファ
 64:         move.w  (a0),L(a1)      *左端のみ
 65:                                 *(x-1,y)の代わりに(x,y)
 66: loop2:  move.w  (a0),C(a1)      *(x,y)
 67:         move.w  (a2),U(a1)      *(x,y-1)
 68:         move.w  DR(a0),R(a1)    *(x+1,y)
 69:         move.w  DD(a0),D(a1)    *(x,y+1)
 70:         bsr     sub             *色を混ぜ合わせる
 71:         move.w  (a0),(a2)+      *下のライン用に覚えておく
 72:         move.w  (a0),L(a1)      *(x-1,y)
 73:         move.w  d0,(a0)+        *1ドット書き込む
 74:         dbra    d4,loop2        *x1-x0-1回繰り返す
 75:
 76:                         *右端の処理
 77:         move.w  (a0),C(a1)      *(x,y)
 78:         move.w  (a2),U(a1)      *(x,y-1)
 79:         move.w  (a0),R(a1)      *(x+1,y)の代わりに(x,y)
 80:         move.w  DD(a0),D(a1)    *(x,y+1)
 81:         bsr     sub             *色を混ぜ合わせる
 82:         move.w  (a0),(a2)+      *下のライン用に覚えておく
 83:         move.w  d0,(a0)+        *1ドット書き込む
 84:
 85:         adda.w  d1,a0           *すぐ下のラインへ
 86:         dbra    d3,loop1        *y1-y0-1回繰り返す
 87:
 88:                         *最下ラインの処理
 89:         move.w  d2,d4           *d4=横ドット数-2
 90:         lea.l   lbuff,a2        *a2=1ラインのバッファ
 91:         move.w  (a0),L(a1)      *左端のみ
 92:                                 *(x-1,y)の代わりに(x,y)
 93: loop3:  move.w  (a0),C(a1)      *(x,y)
 94:         move.w  (a2),U(a1)      *(x,y-1)
 95:         move.w  DR(a0),R(a1)    *(x+1,y)
 96:         move.w  (a0),D(a1)      *(x,y+1)の代わりに(x,y)
 97:         bsr     sub             *色を混ぜ合わせる
 98:         move.w  (a0),L(a1)      *(x-1,y)
 99:         move.w  d0,(a0)+        *1ドット書き込む
100:         dbra    d4,loop3        *x1-x0-1回繰り返す
101:
102:                         *右下隅の処理
103:         move.w  (a0),C(a1)      *(x,y)
104:         move.w  (a2),U(a1)      *(x,y-1)
105:         move.w  (a0),R(a1)      *(x+1,y)の代わりに(x,y)
106:         move.w  (a0),D(a1)      *(x,y+1)の代わりに(x,y)
107:         bsr     sub             *色を混ぜ合わせる
108:         move.w  d0,(a0)         *1ドット書き込む
109:
110: done:   movem.l (sp)+,d0-d7/a0-a2
111:         unlk    a6
112:         rts
113: *
114: sub:
115:         movem.l d1-d4/a1,-(sp)
116:         move.w  (a1)+,d0        *中心の点
117:         DERGB   d0,d1,d2,d3     *RGBごとに
118:         lsl.w   #2,d1           *  4倍
119:         lsl.w   #2,d2           *
120:         lsl.w   #2,d3           *
121:
122:         moveq.l #4-1,d0         *周囲の4点の
123: sublp:  move.w  (a1)+,d4        *  色を
124:         DERGB   d4,d5,d6,d7     *  RGBごとに
125:         add.w   d5,d1           *  加える
126:         add.w   d6,d2           *
127:         add.w   d7,d3           *
128:         dbra    d0,sublp
129:
130:         lsr.w   #3,d1           *8で割る
131:         lsr.w   #3,d2           *
132:         lsr.w   #3,d3           *
133:
134:         RGB     d1,d2,d3,d0     *カラーコードに再構成
135:
136:         movem.l (sp)+,d1-d4/a1
137:         rts
138: *
139:         .bss
140:         .even
141: *
142: lbuff:  .ds.w   GNPIXEL
143:
144:         .end
```

形領域が実画面の端に接しているときに変なアドレスをアクセスしないようにするための処置だ。が，領域が実画面の端に触れていない場合でも同じ処理をするというインチキをやってしまった。このため，常に領域の外枠部分だけボケ具合がゆるい。隣接する領域を別々にボカすようなことをしない限りは目立たないということで目をつぶってもらおう。

輪郭抽出

最後は輪郭抽出だ。色の変わり目（とプログラムで判断した点）を白，それ以外の点を黒で置き換える。リスト8を見てもらおう。リスト7からの変更点のみを示してある。ここで使っている輪郭抽出のアルゴリズムは非常に簡単なもので，“上下左右4ドットの色の平均よりも明るい点を輪郭とみなす”というものだ。数学的にはx方向とy方向に画像を微分する[8]ことだったりするのだが，そんなことはどうでもいいや。リスト8ではこの処理をRGBごとに行い，いずれか1色でも条件を満たしていれば白，そうでなければ黒としている。また，周囲4ドットの平均をとってから中心との差を出すのでは整数除算を行ったときに細部の情報が失われてしまうので，中心の明るさを4倍したものから周囲4ドットの明るさの単純和を引くようにしてある。

もともとこのアルゴリズム自体が画像のノイズに弱い（むしろノイズを強めてしまう）ということもあり，取り込み画像などではそれほどはっきりとした結果が得られるわけではない。“傾向として輪郭らしきものが浮かび上がる”程度だ。あらかじめリスト7などで画像をボカしておけば[9]ある程度は改善されるとはいえ，元絵の傾向によってはプログラムの細部を調整してみるべきかもしれない。たとえば，RGBすべて（ないしは最低2色）について条件を満たした点のみを輪郭とみなすとか，平均よりも一定以上明るくなければならないというように適当なしきい値を設けるなどの手段によってチェックを厳しくすることができる。また，上下左右の4ドットだけではなく斜め方向も考慮するなど，もう少し広い範囲の平均をとるように変更を加えることも考えられる。気がむいたら，いろいろ試してみてもらいたい。

ここで宿題

最後にひさびさの課題を出しておく。今月作ったサブルーチンを参考に，指定された矩形領域に対して以下のような処理を行うサブルーチンを作ってみてもらいたい。珍しいことに解答（例）がリスト9以降に用意してある。

1)　RGB各色成分を半分にするghalftone。：リスト9

2)　リスト4を拡張して任意の色相（青系統とかセピア調とか）で単色化するようなgmonotone_hsv。色はHSVのHとSで色を指定するものとする（HSVからカラーコードへの変換にはIOCSコールのHSVTORGBが使える）。：リスト10

8)　実際にはグラフィック画面の座標や色成分は整数の飛び飛びの値しかとらないから，微分ではなく差分をとっている。

9)　色の平均化はもっとも手軽なノイズ軽減手段だ。

リスト8　GOUTLIN.S

```
  4:         .xdef   goutline
 30: goutline:
114: sub:
115:         movem.l d1-d4/a1,-(sp)
116:         move.w  (a1)+,d0            *中心の点
117:         DERGB   d0,d1,d2,d3         *RGBごとに
118:         lsl.w   #2,d1               *   4倍
119:         lsl.w   #2,d2               *
120:         lsl.w   #2,d3               *
121:
122:         moveq.l #4-1,d0             *周囲の4点の
123: sublp:  move.w  (a1)+,d4            *   色を
124:         DERGB   d4,d5,d6,d7         *  RGBごとに
125:         sub.w   d5,d1               *   減じる
126:         sub.w   d6,d2               *
127:         sub.w   d7,d3               *
128:         dbra    d0,sublp
129:
130:         tst.w   d1                  *RGBいずれか1つでも
131:         bgt     white               *  平均との差が
132:         tst.w   d2                  *  1以上なら白とする
133:         bgt     white               *
134:         tst.w   d3                  *
135:         bgt     white               *
136:
137:         moveq.l #0,d0               *黒
138:         bra     retn
139:
140: white:  move.w  #$ffff,d0           *白
141: retn:   movem.l (sp)+,d1-d4/a1
142:         rts
143: *
144:         .bss
145:         .even
146: *
147: lbuff:  .ds.w   GNPIXEL
148:
149:         .end
```

リスト9　GHALFTONE.S

```
  1:         .include        gconst.h
  2:         .include        gmacro.h
  3: *
  4:         .xdef   ghalftone
  5:         .xref   gramadr
  6:         .xref   gfclip
  7: *
  8:         .offset 0
  9: *
 10: X0:     .ds.w   1
 11: Y0:     .ds.w   1
 12: X1:     .ds.w   1
 13: Y1:     .ds.w   1
 14: *
 15:         .text
 16:         .even
 17: *
 18: ghalftone:
 19: PARPTR = 8
 20:         link    a6,#0
 21:         movem.l d0-d5/a0-a1,-(sp)
 22:
 23:         move.l  PARPTR(a6),a1   *a1=パラメータ受け渡し領域
 24:         movem.w (a1),d0-d3      *d0-d3に座標を取り出す
 25:
 26:         bsr     gfclip          *クリッピングする
 27:         bmi     done            *N=1なら描画の必要なし
 28:
 29:         bsr     gramadr         *左上のG-RAM上のアドレスを得る
 30:
 31:         sub.w   d0,d2           *d2=横ピクセル数-1
 32:         sub.w   d1,d3           *d3=縦ピクセル数-1
 33:
 34:         move.w  #GNBYTE-2,d1    *d1=ライン間のアドレスの差
 35:         sub.w   d2,d1           *(右端から下のラインの
 36:         sub.w   d2,d1           *左端まで)
 37:
 38:         move.w  #%01111_01111_01111_0,d5
 39:                                 *マスクデータ
 40: loop1:  move.w  d2,d4           *d4=横ドット数-1
 41: loop2:  move.w  (a0),d0         *色コードを取り出し
 42:         lsr.w   #1,d0           *   全体を2で割り
 43:         and.w   d5,d0           *   つじつまを合わせる
 44:         move.w  d0,(a0)+        *1ドット書き込む
 45:         dbra    d4,loop2        *繰り返す
 46:         adda.w  d1,a0           *すぐ下のラインへ
 47:         dbra    d3,loop1        *繰り返す
 48:
 49: done:   movem.l (sp)+,d0-d5/a0-a1
 50:         unlk    a6
 51:         rts
 52:
 53:         .end
```

3)　リスト6を任意の色相に拡張したgmonotone_hsy。：2）ができれば簡単だからリストはなし

4)　領域全体に指定色をRGBごとに加えるgaddcolorと色を減じるgsubcolor。：リスト11，12

5)　【自由研究】4）で作ったサブルーチンを利用して疑似的に“減法混色によって色を加える”方法を考えてみよう（色を加えると暗くなるという減法混色の性質と，加法混色系の3原色と減法混色系の3原色が互いに補色の関係にあることを利用する)。

6)　【自由研究】G-RAMには手を加えずパレットの設定のみを変更することで，色反転や量子化，モノクロ化を16色/256色モードで実現してみよう。

7)　【自由研究】リスト8をベースに輪郭を強調するプログラムを作ってみよう。

＊

次回はグラフィックパターンの扱い，いわゆるGET，PUTとか絵の重ね合わせのあたりをつついてみる。で，その次の回は一旦グラフィックを離れてソートとサーチの話，そのまた次の回ではグラフィックに舞い戻ってライン描画と任意多角形の塗り潰しが予定されている。そんなところでまた来月。

参考文献

DAVID F.ROGERS，山口富士夫監修，セイコー電子工業㈱電子機器事業部訳，「実践コンピュータグラフィックス」，日刊工業新聞社

リスト10　GMONOHSV.S

```
 1:         .include        iocscall.mac
 2:         .include        gconst.h
 3:         .include        gmacro.h
 4: *
 5:         .xdef   gmonotone_hsv
 6:         .xref   gramadr
 7:         .xref   gfclip
 8: *
 9:         .offset 0
10: *
11: X0:     .ds.w   1
12: Y0:     .ds.w   1
13: X1:     .ds.w   1
14: Y1:     .ds.w   1
15: H:      .ds.w   1
16: S:      .ds.w   1
17: *
18:         .text
19:         .even
20: *
21: gmonotone_hsv:
22: PARPTR = 8
23: TBLSIZ = RGBGRAD*2
24:         link    a6,#-TBLSIZ
25:         movem.l d0-d7/a0-a1,-(sp)
26:
27:         move.l  PARPTR(a6),a1   *a1=パラメータ受け渡し領域
28:         movem.w (a1),d0-d3      *d0-d3に座標を取り出す
29:
30:         bsr     gfclip          *クリッピングする
31:         bne     done            *Z=0なら描画の必要なし
32:
33:         bsr     gramadr         *G-RAM上のアドレスを得る
34:
35:         sub.w   d0,d2           *d2=横ドット数-1
36:         sub.w   d1,d3           *d3=縦ドット数-1
37:
38:         move.l  H(a1),d1        *hsv(H,S,0)～hsv(H,S,31)の
39:         lea.l   -TBLSIZ(a6),a1  * 色テーブルを作っておく
40:         lsl.w   #8,d1           *
41:         moveq.l #RGBGRAD-1,d4   *
42: loop0:  IOCS    _HSVTORGB       *
43:         move.w  d0,(a1)+        *
44:         addq.b  #1,d1           *
45:         dbra    d4,loop0        *
46:
47:         move.w  #GNBYTE-2,d1    *d1=ライン間のアドレスの差
48:         sub.w   d2,d1           *（右端から下のラインの
49:         sub.w   d2,d1           *左端まで）
50:
51:         lea.l   -TBLSIZ(a6),a1
52: loop1:  move.w  d2,d4           *d4=横ドット数-1
53: loop2:  move.w  (a0),d0         *色コードを取り出し
54:         DERGB   d0,d5,d6,d7     *RGBに分解する
55:         MAX     d6,d5           *R,G,Bの最大値を
56:         MAX     d7,d5           *  d5に求め
57:         add.w   d5,d5           *hsv(H,S,d5)の色で
58:         move.w  0(a1,d5),(a0)+  *  点を打つ
59:         dbra    d4,loop2        *繰り返す
60:         adda.w  d1,a0           *すぐ下のラインへ
61:         dbra    d3,loop1        *繰り返す
62:
63: done:   movem.l (sp)+,d0-d7/a0-a1
64:         unlk    a6
65:         rts
66:
67:         .end
```

リスト11　GADDCOLOR.S

```
 1:         .include        gconst.h
 2:         .include        gmacro.h
 3: *
 4:         .xdef   gaddcolor
 5:         .xref   gramadr
 6:         .xref   gfclip
 7: *
 8:         .offset 0
 9: *
10: X0:     .ds.w   1
11: Y0:     .ds.w   1
12: X1:     .ds.w   1
13: Y1:     .ds.w   1
14: COL:    .ds.w   1
15: *
16:         .text
17:         .even
18: *
19: gaddcolor:
20: PARPTR = 8
21:         link    a6,#0
22:         movem.l d0-d7/a0-a1,-(sp)
23:
24:         move.l  PARPTR(a6),a1   *a1=パラメータ受け渡し領域
25:         movem.w (a1),d0-d3      *d0-d3に座標を取り出す
26:
27:         bsr     gfclip          *クリッピングする
28:         bmi     done            *N=1なら描画の必要なし
29:
30:         bsr     gramadr         *左上のG-RAM上のアドレスを得る
31:
32:         sub.w   d0,d2           *d2=横ピクセル数-1
33:         sub.w   d1,d3           *d3=縦ピクセル数-1
34:
35:         move.w  #GNBYTE-2,d1    *d1=ライン間のアドレスの差
36:         sub.w   d2,d1           *（右端から下のラインの
37:         sub.w   d2,d1           *左端まで）
38:
39:         move.w  COL(a1),d4      *d0=加える色
40:         DERGB   d4,d5,d6,d7     *
41:
42: loop1:  move.w  d2,d4
43:         swap.w  d1              *レジスタが足りないから
44:         swap.w  d2              *  データレジスタの
45:         swap.w  d3              *  上位ワードも使う
46: loop2:  move.w  (a0),d0         *色コードを取り出し
47:         DERGB   d0,d1,d2,d3     *  RGBごとに
48:         add.w   d5,d1           *   色成分を加える
49:         add.w   d6,d2           *
50:         add.w   d7,d3           *
51:         MIN     #RGBMAX,d1      *ただし32以上になったら
52:         MIN     #RGBMAX,d2      *  31に修正する
53:         MIN     #RGBMAX,d3      *
54:         RGB     d1,d2,d3,d0     *カラーコードを得る
55:         move.w  d0,(a0)+        *1ドット書込む
56:         dbra    d4,loop2
57:         swap.w  d1
58:         swap.w  d2
59:         swap.w  d3
60:         adda.w  d1,a0           *すぐ下のラインへ
61:         dbra    d3,loop1        *d3回繰り返す
62:
63: done:   movem.l (sp)+,d0-d7/a0-a1
64:         unlk    a6
65:         rts
66:
67:         .end
```

リスト12　GSUBCOLOR.S(リスト11からの変更点のみ)

```
 4:         .xdef   gsubcolor
19: gsubcolor:
46: loop2:  move.w  (a0),d0         *色コードを取り出し
47:         DERGB   d0,d1,d2,d3     *  RGBごとに
48:         sub.w   d5,d1           *   色成分を加える
49:         sub.w   d6,d2           *
50:         sub.w   d7,d3           *
51:         MAX     #0,d1           *ただし0未満になったら
52:         MAX     #0,d2           *  0に修正する
53:         MAX     #0,d3           *
54:         RGB     d1,d2,d3,d0     *カラーコードを得る
55:         move.w  d0,(a0)+        *1ドット書込む
56:         dbra    d4,loop2
```

ひとり占いTEN

Iketani Masahiko 池谷 昌彦

HEARTの作者，池谷さんのカードゲーム第2弾です。このシリーズでは初めてのひとり遊び，簡単なリストですから気軽に入力さてください。なお，このプログラムの実行には3月号に掲載，6月号に収録されたCARD. FNCが必要です。

ひとり占い

このゲームはもともとトランプのひとり遊びとなっています。ですから対戦相手やスコア，思考ルーチンなどはありません。たいていのひとり遊びと同様，このゲームもカードをシャッフルした時点ですでに結果は確定しています。人間の作業はその結果を確認していくことだけなのです。偶然性だけでゲームが成立しているわけですが，逆にいえばそのゲームの出来不出来によって占いのような意味で使うこともできるわけです。

ルール解説

まず，場には5枚のカードを横2列，その下に3枚，計13枚のカードを表向きに並べます。このなかで，足して10になる2枚のカード，または同じスートの10,J,Q,Kをまとめて場から取り除いていきます。なくなったところには手札からカードを加えていきます。

このようにして，すべてのカードを取り除くことができれば，どんな願いもかなうといいます。残った手札が10枚以内なら他人の援助や協力を得て願いごとがかなえられるでしょう。あまり多く残ったときは，障害が多くなり油断すると思わぬ災難にあいます。

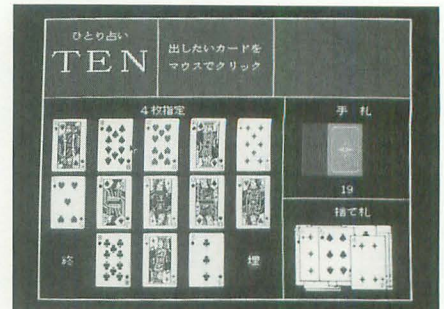
操作方法

プログラムを起動すると画面に場札を表示していきます。そのなかから，取り除けるものを選んでマウスクリックしてください（左右どちらでもかまいません）。選んだ2枚のカードがちゃんと取り除けるカードならそれらのカードが画面から消えます。10,J,Q,Kのカードは4枚単位でしか取り除けません。

場札の欠けた部分は「埋」のボタンをクリックすることで補充されます。

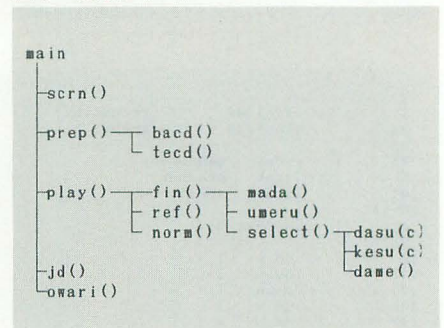
どうしても取れなくなった場合は，「終」のボタンをクリックしてください。見落としがないかどうかをコンピュータがチェックしたうえで，残りのカード数を表示します。もう一度やる場合はリプレイの選択で「Y」のボタンを，もうやめたいというときは「N」のボタンをクリックします。

＊　＊　＊

プログラムはなるべく簡単に，短く，要領よくと思って作ったのですが，実力不足で10Kバイトを超えてしまいました。BASICの場合，行番号だけでも馬鹿にならないなあと，いまさらながら驚いています。

なお，CARD.FNCのサンプル“99”の捨て札の表示法に感心しましたので，今回アイデアを借用させていただきました。

プログラムで使われている関数を図1にまとめてみました。カードゲームを作りたいという方は参考にしてください。

図1　関数チャート

```
main
 ├scrn()
 ├prep()─┬ bacd()
 │        └ tecd()
 ├play()─┬fin()─┬ mada()
 │        ├ref() ├ umeru()
 │        └norm()└ select()─┬dasu(c)
 │                            ├kesu(c)
 │                            └dame()
 ├jd()
 └owari()
```

リスト1　TEN.BAS

```
 10 /*
 20 /* TEN
 30 /*   Programmed by M.I.,Jul.20,'90
 40 /*
 50 screen 1,1,1,1:console ,,0
 60 int flag=0
 70 dim int cc(51),ba(12),te(38),m(2)
 80 palet(1,0)
 90 /* main program
100 while flag=0
110    scrn()
120    prep()
130    play()
140    jd()
150 endwhile
160 owari()
170 end
180 /* preparation of screen
190 func scrn()
200    int i
210    vpage(0)
220    apage(1):fill(0,0,511,511,0)
230    apage(2):fill(0,0,511,511,0)
240    fill(0,0,511,144,10):fill(0,145,511,511,8)
250    box(0,0,511,511,15):box(1,1,510,510,15)
260    line(2,144,509,144,15):line(160,2,160,143,15)
270    line(320,2,320,143,15):line(336,144,336,509,15)
280    line(337,328,509,328,15)
290    symbol(40,24,"ひとり占い",1,1,1,15,0)
300    symbol(10,58,"ＴＥＮ",2,2,2,1,0)
310    symbol(8,56,"ＴＥＮ",2,2,2,15,0)
320    symbol(408,160,"手　札",1,1,1,15,0)
330    symbol(408,344,"捨て札",1,1,1,15,0)
340    vpage(6)
350 endfunc
360 /* preparation
370 func prep()
380    int i,j,a,b,k
390 /* music data set
400    m_init()
410    for i=1 to 8:m_alloc(i,2000):m_assign(i,i):next
420    m_trk(1,"q2@23v13o3t200e64")
430    m_trk(2,"q8@15v13o3t100c4")
440    m_trk(3,"q6@56v14o5t100l16ae")
450    m_trk(4,"q8@57v13o3t100c8")
460    m_trk(5,"q8@57v13o3t100e8")
470    m_trk(6,"q4@57v13o3t100g8")
480    m_trk(7,"q8@57v13o4t100c8")
490    m_trk(8,"q8@52v15o4t80 c2")
500 /* deal
```

```
 510    randomize(val(mid$(time$,4,2)+right$(time$,2)))
 520    for i=0 to 51:cc(i)=i+1:next
 530    m(0)=13:m(1)=39:m(2)=0
 540    for i=0 to 99
 550       a=int(rnd()*52):b=int(rnd()*52)
 560       k=cc(a):cc(a)=cc(b):cc(b)=k
 570    next
 580    for i=0 to 12:ba(i)=cc(i):next
 590    for i=13 to 51:te(i-13)=cc(i):next
 600    for i=0 to 51:cc(i)=0:next
 610    apage(1)
 620    bacd():tecd()
 630    circle(296,440,16,4,,,350):paint(296,432,6)
 640    symbol(288,432,"埋",1,1,1,15,0)
 650    circle(40,440,16,4,,,350):paint(40,432,4)
 660    symbol(33,432,"終",1,1,1,15,0)
 670 endfunc
 680 /* play
 690 func play()
 700    int ss,q,g=0
 710    while g=0
 720       if m(0)+m(1)>0 then q=select() else q=20
 730       switch q
 740          case 20:ss=fin():break
 750          case 15:ref():break
 760          default:norm(q)
 770       endswitch
 780       if ss=1 then g=1
 790    endwhile
 800 endfunc
 810 /*
 820 func norm(a)
 830    int b,c,d,a1,b1,c1,d1,a2,b2,c2,d2
 840    a1=(ba(a)-1) mod 13:a2=(ba(a)-1)¥13
 850    if a1<9 then {
 860       b=select()
 870       if a<>b and b<13 then b1=(ba(b)-1) mod 13
 880       if a1+b1=8 then {
 890          dasu(a):m_play(4):dasu(b):m_play(5):return()
 900       } else {
 910          kesu(a):kesu(b):dame():return()}
 920    } else if a1>8 then {
 930       symbol(136,160,"４枚指定",1,1,1,15,0)
 940       b=select()
 950       if a<>b and b<13 then b1=(ba(b)-1) mod 13:b2=(ba(b)-
1)¥13
 960       if b1>8 and b2=a2 then {
 970          c=select()
 980          if c<>b and c<13 then c1=(ba(c)-1) mod 13:c2=(ba(
c)-1)¥13
 990       } else {
1000          kesu(a):kesu(b):dame()
1010          fill(136,160,200,176,0):return()}
1020       if c1>8 and c2=a2 then {
1030          d=select()
1040          if d<>c and d<13 then d1=(ba(d)-1) mod 13:d2=(ba(
d)-1)¥13
1050       } else {
1060          kesu(a):kesu(b):kesu(c):dame()
1070          fill(136,160,200,176,0):return()}
1080       if d1>8 and d2=a2 then {
1090          dasu(a):m_play(4):dasu(b):m_play(5)
1100          dasu(c):m_play(6):dasu(d):m_play(7)
1110          fill(136,160,200,176,0)
1120       } else {
1130          kesu(a):kesu(b):kesu(c):kesu(d):dame()
1140          fill(136,160,200,176,0)}
1150    }
1160 endfunc
1170 /*
1180 func fin()
1190    int i,j,e,k=0,m1=0,m2=0,m3=0,m4=0,n=0
1200    if m(0)<12 and m(1)>0 then e=0:mada():return(e)
1210    for i=0 to 11:for j=i+1 to 12
1220       if ba(i)=0 or ba(j)=0 then continue
1230       if (ba(i)-1) mod 13+(ba(j)-1) mod 13=8 then k=1
1240    next:next
1250    for i=0 to 12
1260       if ba(i)>=10 and ba(i)<=13 then m1=m1+1
1270       if ba(i)>=23 and ba(i)<=26 then m2=m2+1
1280       if ba(i)>=36 and ba(i)<=39 then m3=m3+1
1290       if ba(i)>=49 and ba(i)<=52 then m4=m4+1
1300    next
1310    if m1=4 or m2=4 or m3=4 or m4=4 then n=1
1320    if k+n>=1 then e=0:mada():return(e)
1330    if k+n=0 then e=1:return(e)
1340 endfunc
1350 /*
1360 func ref()
1370    if m(0)<12 and m(1)>0 then umeru()
1380 endfunc
1390 /* judge
1400 func jd()
1410    int x,y,bl,br,ten
1420    fill(321,2,509,143,12)
1430    ten=m(0)+m(1):m_play(8)
1440    if ten=0 then symbol(325,48,"おめでとう！",2,2,1,15,0)
         else {
1450       symbol(344,48,str$(ten)+"枚　残りました",1,1,1,15,0
)}
1460    symbol(336,96,"もう１度？",1,1,1,15,0)
1470    fill(432,96,463,112,15):symbol(444,96,"Y",1,1,1,1,0)
1480    fill(465,96,496,112,15):symbol(476,96,"N",1,1,1,1,0)
1490    mouse(1)
1500    msarea(433,96,495,111)
1510    repeat
1520       msstat(x,y,bl,br)
1530    until bl=-1 or br=-1
1540    mspos(x,y)
1550    mouse(0)
1560    if x>465 then flag=1
1570 endfunc
1580 /* owari()
1590 func owari()
1600    vpage(2):apage(1)
1610    fill(0,0,511,511,2)
1620    symbol(272,400,"お疲れ様でした",1,1,2,15,0)
1630 endfunc
1640 /*
1650 func select()
1660    int i,x,y,bl,br
1670    symbol(176,48,"出したいカードを",1,1,1,15,0)
1680    symbol(176,80,"マウスでクリック",1,1,1,15,0)
1690    mouse(1)
1700    msarea(17,184,327,487)
1710    setmspos(168,336)
1720    repeat
1730       msstat(x,y,bl,br)
1740    until bl=-1 or br=-1
1750    mspos(x,y)
1760    mouse(0)
1770    erupms()
1780    if y<279 and (x>16 and x<328) then {
1790       i=(x-16)¥64:box(15+i*64,183,63+i*64,280,5)
1800    } else if (y>288 and y<384) and (x>16 and x<328) then {
1810       i=(x-16)¥64+5:box(15+(i-5)*64,287,63+(i-5)*64,384,5)

1820    } else if y>392 and (x>80 and x<256) then {
1830       i=(x-80)¥64+10:box(79+(i-10)*64,391,127+(i-10)*64,48
8,5)
1840    } else if (y>428 and y<456) and (x>280 and x<312) then
{
1850       i=15
1860    } else if (y>428 and y<456) and (x>24 and x<56) then i=
20 else select()
1870    return(i)
1880 endfunc
1890 /*
1900 func kesu(c)
1910    int x,y
1920    if c<10 then x=(c mod 5)*64+15 else x=(c mod 5)*64+79
1930    y=(c ¥ 5)*104+183
1940    box(x,y,x+48,y+97,0)
1950 endfunc
1960 /*
1970 func dasu(c)
1980    int x,y,s,t
1990    if c<10 then x=(c mod 5)*64+15 else x=(c mod 5)*64+79
2000    y=(c ¥ 5)*104+183
2010    fill(x,y,x+50,y+98,0)
2020    m(0)=m(0)-1:if m(0)<0 then m(0)=0
2030    m(2)=m(2)+1
2040    s=int(rnd()*96)+352:t=int(rnd()*24)+376
2050    c_put(s,t,ba(c)):box(s-1,t-1,s+47,t+96,8)
2060    ba(c)=0:wait(10)
2070 endfunc
2080 /*
2090 func umeru()
2100    int i,x,y
2110    for i=0 to 12
2120       if ba(i)=0 then {
2130          if i<10 then x=(i mod 5)*64+16 else x=(i mod 5)*6
4+80
2140          y=(i ¥ 5)*104+184
2150          if m(1)=0 then fill(360,182,408,288,0):break
2160          c_put(x,y,te(m(1)-1)):m_play(3)
2170          ba(i)=te(m(1)-1)
2180          fill(m(1)*2+360,182,m(1)*2+408,288,0)
2190          c_put(m(1)*2+358,192,0)
2200          m(0)=m(0)+1
2210          m(1)=m(1)-1:if m(1)<0 then m(1)=0
2220       }
2230    next
2240    fill(416,304,432,320,0)
2250    symbol(416,304,str$(m(1)),1,1,1,15,0)
2260 endfunc
2270 /*
2280 func dame()
2290    symbol(180,48,"出せません",1,1,2,5,0)
2300    m_play(2):wait(50):erupms()
2310 endfunc
2320 /*
2330 func mada()
2340    symbol(168,48,"終ってません",1,1,2,5,0)
2350    m_play(2):wait(50):erupms()
2360 endfunc
2370 /* ba card
2380 func bacd()
2390    int i
2400    for i=0 to 4
2410       c_put(i*64+16,184,ba(i)):m_play(1)
2420       c_put(i*64+16,288,ba(i+5)):m_play(1)
2430    next
2440    for i=0 to 2
2450       c_put(i*64+80,392,ba(i+10)):m_play(1)
2460    next
2470 endfunc
2480 /*
2490 func tecd()
2500    int i
2510    for i=0 to m(1)-1
2520       c_put(i*2+362,192,0)
2530       line(i*2+362,192,i*2+362,288,8):m_play(1)
2540    next
2550    fill(416,304,432,320,0)
2560    symbol(416,304,str$(m(1)),1,1,1,15,0)
2570 endfunc
2580 func wait(t)
2590    int i
2600    for i=0 to t*100:next
2610 endfunc
2620 /*
2630 func erupms()
2640    fill(161,3,319,143,0)
2650 endfunc
```

X68000

住所録あれこれ

Ogikubo Kei 荻窪 圭

住所録といってもバカにしてはいけません。パーソナルユーザーにとってのデータベースのエッセンスはここにあります。今回はいろいろなタイプのソフトを使ってデータを使いやすいかたちで利用するためのポイントを探ってみましょう。

住所録からいこう

実をいうと，私のハードディスクはたいへん無駄な状況になっている。いろんなソフト用のフォーマットで同じ（ような）内容のデータベースがいくつもあるのだ。

CYBERNOTE PRO-68K用住所録，CARD PRO-68K用住所録，Kamikazeのデータベースフォーマットの住所録,Kamikazeの表計算シートに書かれた住所録,Stationery PRO-68Kの住所録，そして，X68000じゃないけど電子手帳に入っている住所録。いまは電子手帳中心にしているから，CYBERNOTEのデータベースに入っているのが最新バージョンである。

最初に作られたのが，Kamikaze用データベースだ。きっかけは，電話番号簿をリフィルに印刷して，システム手帳に挟みたい！　だった。手書きだと削除・挿入・変更が面倒で，システム手帳が厚くなってしまうからね。

で,Kamikazeのデータベースに打ち込んで，表計算シートに名前と電話番号をコピーして，試行錯誤しながらリフィルに印刷したのだ。おかげで快適なシステム手帳ライフを送らせていただいた。手書きの汚い文字より読みやすい24ドット文字というのはありがたいものである。

その後,Stationery PRO-68Kのレビューを書くために電子手帳に手を出したのが間違いだった。

まず,KamikazeからCSV形式(Kamikazeでは標準テキスト形式と呼んでいる)で項目の順番を合わせてセーブし，それをStationery用アドレスファイルとして指定し(StationeryはCSV形式をそのまま使う)，電子手帳にダウンロード。当時は2Mバイトのメモリがあればなんか広大な空間を味わえるような幸せな時代だったので(たった1年前！)，Stationeryが300Kバイトほど占有したとて許せた。そんでもって，カレンダーを見たり気紛れで電話したりするのに重宝した。

ところが，CYBERNOTE PRO-68Kのレビューをすることになって，CYBERNOTEのコンバート機能でCSV形式をCYBERNOTEのデータベース形式にコンバート。ここでStationeryは使われなくなる。

CYBERNOTEのほうがマウスベースで気軽に使えたし，電子手帳とのインタフェイスがしっかりしてたから。だが問題点は，立ち上がりに時間がかかることと，いちいち使用するファイルを指定せねばならないことだった。あっ，と思ったときにすぐ立ち上がる機動力と，チャイルドプロセスでも使えるコンパクトさが現代に生きる個人データベースのポイントなのだ。

そんでもって，どうせなら，ちゃんと項目分けをして本当にほしいデータを集めたデータベースを作ろう，と，CARD PRO-68Kに手を出したはいいが……。

住所録は何で作るのがいいか

私は上の経験を通じてさまざまなことを学んだ。「人生に必要な知恵はすべて幼稚園の砂場で学んだ」人がいるらしいくらいだから，私がここで某かを学んだとしてもバチは当たらないだろう。いつでもどこでも学ぶのが大人の基本である。

で，結論をいおう。住所録に使うソフトは，ワープロだってエディタだって，なんだって，それでいいのだ。

この悟りは重要かもしれない。「住所録はデータベースだから，データベースソフトを使おう」などという言葉尻に踊らされていると，電通にコロリとだまされて，馬鹿な出費を重ねる善良な市民になるのだから。要はなんだっていいのである。数十件程度なら標準ワープロのほうが安くすんで（安いどころか無料！）手軽だ。

しかし，どの方式にも長所短所がある。それを追いかけつつ，データベースの本質に迫りたい（これはいつもの大風呂敷)。

十人十色な用途

パソコンで住所録を管理するとき，なにが優れているか，というのを考えてみる。簡単だね。変更が楽。追加・削除も楽。印刷すれば手書きの暗号文字より綺麗だし，楽。データが増えても検索できる。

しかし，大いなる欠点もある。データを手で打ち込まなければならない，こと。手で打ち込む労力に対する見返りがほしい。

1)　システム手帳のリフィルに印刷する。これはとても重宝する。なぜなら，手書きだと分類しようと思うと空白だらけのリフィルになって手帳がすぐ厚くなるし，なにも考えないで書き込むと検索が異様に面倒だ。必要な人だけ必要な情報だけ印刷できればシステム手帳が分厚くならないのだ。昨今，会社が引っ越したとか，転勤したとかで，年に何人かは連絡先が変わる。そんなときなにかと便利である。

2)　電子手帳で使う。いうまでもなく，電子手帳の周辺機器としてパソコンはとても優秀である。

3)　人間，年を取るといろんな知り合いができるので，分類したい。高校時代の同級生と会社の同僚とわけのわかんない知り合いとOh!X編集部の電話番号なんかがごっちゃになっていると使いづらいし，50音順なんかでは解決がつかない。そんなとき，分類コードをつけて分類するとものすごく便利である。

大学時代の知り合いの女の子で独り暮らし，という条件で検索して，そのなかから遊んでくれる相手を捜す，なんてのは冗談だが，年賀状を誰に出そうか，とか，引っ越し案内を誰に出そうか，とか，結婚式の招待状を誰に出そうか，というとき，いちいち名簿や古いアドレス帳を持ち出すのはとても面倒だ。で，とにもかくにも知り合いが全部放り込んであるデータベースから抽出するのだ。

4)　各種虚礼の際，いちいち手で住所を書

くのがうっとーしかったりすると，パソコン(&プリンタ) に任せることができる。
5) お仕事なんかだと，相手先のいろんな情報はまとめておきたい。
6) パソコンは手帳と違って，なくしたり，どこかへ紛れ込んだりしない。フロッピーをなくすと困るが，まあ，それは例外だ。
7) 自分の友達の数を確認したり，“君のためなら死ねる”リストを作ったりといろいろ遊んで，アドレス帳おたくを味わう。
8) その他，あまたのビジネス誌やパソコン入門をひょーぼうするガイド本が語る綺麗ごとの数々。

さあ，これを全部満たそうと思うと，大変だ。電子手帳の使えるソフトは任意の分類ができないぞ。リフィル印刷をしたいだけでデータベースソフトを買うのはばかばかしいぞ。というわけで次をどうぞ。

1：付属ワープロで作る

なんといっても，ただである。さらに，罫線や修飾が豊富なので，リフィルに印刷すると綺麗だ。データベースソフトのように項目名やその長さなんていう面倒な設定(データベースの設計) がいらない。

と，いうわけで，数十件程度のデータならワープロで十分である。項目の長さを決めなくてもいい (長い会社名のやつは2行にするとかができる)。人の扱うデータはたいてい非定型だからだ。

たとえば，第1章を元データにし，第2章をリフィル印刷用にする。それで，1章から印字したいデータだけをコピー&ペーストして使う。

難をいえば，ソートがない。データが増えると管理が面倒，という欠点がある。データの桁数を変えるといったことはあまり考えたくない。しかし，ワープロで十分だというケースは多いのではないか。

2：Hypewordを使う

付属ワープロよりも，Hyperwordは階層構造で抜きん出ている。50音ならば，先頭シートに“あ”から“ん”とアルファベットなんかを入れて，それをクリックして開く子シートにそれぞれ名前や電話番号が書いてある，てな寸法だ。検索に便利だね。そんでもって，さらに子シートを開いて，手紙の管理，なんていうアワワなことにも使えたりする。

秀逸なのが，Hyperwordのタブ機能。ワープロみたいにタブをスペースに変換せず，タブコードで持っている。タブ位置を設定し直すと，文書全部のタブを新しく設定した位置に直してくれるのだ。

だから，表を作るとき，あとから項目位置を直すのに非常に便利。付属ワープロより優秀な点のひとつだろう。

問題点は2つ。立ち上がりに時間がかかることと，Hyperwordはその性格上独自のフォーマットでデータを持っているために世界が閉じていること。クローズドワールド！ Hyperwordで作った複雑な階層構造を持ったデータをほかのソフトにコンバートしようと思うととても面倒だ。

3：エディタを使う

エディタを使うメリットは，1にも2にも軽い (プログラムサイズが小さい!)ことである。すぐ立ち上がる。チャイルドプロセスからでも立ち上がる。

項目間はタブコードでも突っ込んでおけばいい。揃っていればなんとかなる。データの抽出をしたかったら，FINDコマンドがあるではないか (ちょっと面倒だけど)。味気なくてもよければ，エディタだって十分データベースに使えたりするのである。

パソコン通信マニアなら，エディタを使って，よくメールを送る人間のホスト名とハンドルネーム，IDや自己紹介などを入れたテキストファイルを作っておく。通信ソフトからチャイルドプロセスでエディタを立ち上げてメール先確認をしたり，チャット相手を確認したりに役立つだろう。

ちょっと必要なときには欠かせない。

＊

以上の3つは，もともとテキストを書くための道具である。よって，1レコードが長くなった (1人のデータにいろんな情報を突っ込みたい) ときに不便だ。名前と住所と電話番号 (2つ3つはほしい) くらいならいいが，加えて出身校やどこで知り会ったかという分類や，最近連絡を取ったのはいつかなど付随する情報をたくさんつけたい場合に困る。さらに，タックシール印刷などいかにもパソコンらしいというか大人じゃん(?)的な仕事をさせようと思ったとき，テキスト系ツールは致命的である。

4：Kamikazeを使う

住所録程度のデータベースなら，単なる表にすぎない。Kamikazeのように“表形式のデータを処理するソフト”(あえて表計算とはいわないよ) はとても便利である。

Kamikazeをデータベースとして使う方法は2つある。ひとつはワークシートをそのまま表の形で使うこと。もうひとつは独自のデータベース機能を使うことである。

表計算でデータベースするときのメリットはいろいろあるが，
a) 項目の長さがいつでもマウスでぴよ〜んと変更できる。
b) 項目の追加，削除が簡単にできる。
c) ソートが簡単。
というところがあげられよう。

ただ，表の上にでででっとデータベースを作ると，検索なんかが面倒になる。メニューバーから検索を選択して実行しよう。

私はデータベース機能を使っていた。まず，表にでででれーっと必要な住所録を書き込み，連続登録で一気にデータベースへと書き込んだ。目的はリフィルへの印字だったので，目的に応じて検索してはワークシートへとペーストし，整形して印字する。もちろん，リフィル印字用のワークシートは作っておくのだ。

Kamikazeのデータベースの欠点として，追加やデータの変更，検索条件の入力という3つの面倒があげられる。検索くらいもっと対話的にしてほしいものである。

ワープロ系ツールで不可能なことがKamikazeでは可能になる。“分類”である。

たとえば，男か女かのセル (SEXセル) を作り，データ全体を範囲指定して，SEXセルをキーにソートすると，ほうら，男と女が綺麗に分離できた。

こんなことがなんの役に立つかというと，会社別に分類したり，出身校別に分離したり，そのほかいろいろできると便利なことがあるのだ。

ついでに，いやらしいことだが，年賀状を出したか来たかを入れておくと，毎年毎年の虚礼データが出来上がる。

＊

続いて，データベース専用ソフトを使う場合を見てみよう。大人風にいうと，DBMS，データベースマネジメントシステム，っていうやつだ。

データベースソフトはどんなときに使うと便利か。という問題がある。実のところ，個人の個人による個人のためのデータベースに専用ソフトは必要か，ということに私は疑問を抱いているのだ。

データベース専用ソフトがほかより優れているのは，私が見る限り，次の点によってである。
a) 画面設計
b) 凝った印刷

c)　確実なデータ管理
d)　多彩なデータ検索
a)だが，データベースソフトのみが，1件につき1画面という構成でデータを管理できる。住所録や電話帳程度なら1画面に1件しか見られないというのはうっとうしいだけだが，1件につき大量のデータを放り込みたい場合に自由な画面設計というのは有用だ。また，住所録の場合でも，名前と住所だけとか，名前と電話番号だけといった必要な項目のみの抽出ができるといったメリットは欠かせない。
b)へいくと，もう，データベース万歳の世界で，CARD PROの場合，リフィル印刷用フォーム集が出ていたりする。
c)っていうと，“不用意なデータ書き換えや削除”が起きにくいということだ。
d)なんてのはもういわずもがなで，データベースソフトなんて使おうという人は，データを蓄えておくことと同時に，上手に引き出せることが目的なのだ。いかに早く検索，抽出するかが勝負だ。たいていの場合，高速検索するためのキー項目と，そーでない項目がある。ちょっと面倒でもandとかorを駆使してやれば，何千件もあるなかから必要なものを引っ張り出してきてくれるわけ。「お,OB会の幹事になっちまったぜい」てなとき，条件にはまる人だけを抽出して住所と名前をハガキ印刷，てなことができるのはデータベースソフトだけだ。
なんといってもデータベースはデータが大量になったとき，どう管理するか，いかに素早く検索・抽出するかが勝負なの。
面倒なのがデータの入力。1件ごとにデータをセーブするので，遅い。それはなぜかというと，データベースだからだ。データの管理を安全・確実にやろうと思うと，その都度書き込むに限る。
そんでもって，X68000の場合，汎用データベースのCARD PROのほか，特定用途用データベースが2本あるわけだ。その2本がStationeryとCYBERNOTEだ。具体的にいこう。

5：Stationeryってか

私が思うに，いままで家庭内対象のツールが発達しなかったのは，立ち上がりに時間がかかるからである。さて，誰々さんちに暑中見舞い書かなきゃだわ，ってなとき，よっこいしょっとパソコンの電源を入れ，ソフトを立ち上げ，ファイルを選択して，検索する，ってな面倒なことを誰がするだろうか。私はしない。
世の中には手間をかける価値のある仕事とそうでない仕事があって，何千件のなかから都内に住むお得意さんを検索したいときに手間をかけるのはかまわないが，ちょっと友達の電話番号を知りたいときに何分もかけて検索したい人はいないのだ。
んでもって，どーすれば速いかと考えると，メモリ上に常駐すればいいわけだ。シュタっと呼び出せるわけだ。たとえ300Kバイトほどメモリが減ろうとも，メモリなんて金で買える程度のものではないか。
つーわけで，付加価値として電子手帳とのリンクを用意してStationeryは出てきたのである。
しかし，中途半端な感は免れない。電子手帳のサブセットみたいな機能なのだから。さらに，X68000のスペックを考えると無謀に近い試みだったことがわかる。テキスト画面を待避させてからStationeryは顔を出すのだが，待避ってもX68000のテキスト画面は512Kバイトもあるのだ。うち，使われていないと思われる（1024×1024−764×512）ドット分を抜いたとしても約380Kバイト，0,1プレーンのみとしても190Kバイト分待避しなければならないわけで，メモリを食うわけである。
こいつの生き残る道は2つだ。機能の充実か機能の分散か。充実させるなら，項目を増やすなり電卓をつけるなりいろいろする必要がある。分散は，機能を全部バラバラにして，必要なものだけ常駐させるようにする。
私の望みは，機能を分散させて，SX-WINDOW対応にすること。これはいいでっせ。SX-WINDOWだったら，住所録ウィンドウを出しといて，いつでもペロっと必要なデータをカット&ペーストできるのだから。
というわけで，Stationery PRO SXを待望しつつ次である。

6：CYBERNOTE PRO-68Kを使う

CYBERNOTE PRO-68Kは，住所録とスケジューリング用ソフトである。中身はCARD PRO-68Kだが，ユーザーインタフェイスを変えてあるため，まったく異なったソフトの印象だ。
データベースソフトの設計の面倒臭さを肩代わりしてくれる代わりに自分で設計できない硬さを背負っている。
たとえば，“勤務先”を入れるところがないとか，“ユーザー項目”がないなど。
立ち上がるのに時間がかかるため，とっさに使いたいときにイライラする。世界時計は使わない人は使わないのだから，外せるようにしてほしいものである。
笑えるのが顔写真コーナーだ（NECの電子手帳ETのモンタージュ機能も同じ）。しかし，あそこにちょっとした絵を描けるグラフィックツールがついていたらそんなに間抜けではなくなる。小さな地図なんかを入れることができるからだ。名刺管理とかいうほど仕事で使うのなら，簡単な地図が必要なことはいろいろある。リフィルに印刷して持ち歩くとき，便利だ。
CYBERNOTEで充実しているのはデータコンバート機能と家計簿だろう。そんでもって，電子手帳を使いたいならやはりこの手のソフトが1本必要だ。どうせなら，CYBERNOTEで使っている電子手帳用デバイスドライバは公開してシステムディスクにでもつけてくれると，ありがたい人は多かろう。

7：CARD PROを使う

住所録とひと口にいうが，必要な情報というのは人それぞれによって違う。対象によっても違う。
私は薄情で怠慢だから，住所なんてたいてい使わない。年賀状だってできたら廃止したいくらいだ。しかし，電話番号はたくさん必要だ。
あるギョーカイ人の友達を捕まえようと思ったら，自宅にかけて，勤務先にかけて，(時には帰省先にかけて)，そいつの彼女のアパートにかけて，それでもいなかったら2号さんのとこにかけて，と，多いときで5カ所にかける必要があるのだ。
ほうら，電話番号はたくさん必要でしょ。通信機器に囲まれた現代人は1人当たりいくつ電話番号があっても不思議はない。なのに，住所録ソフトやら市販のアドレス帳なんてのは電話がひとつFAXがひとつってな石器時代構成になっていて感覚を疑う。
閑話休題。データベースのデータというのはなかなか思いどおりに並んではくれない。ジャンル別にファイルを分けるのもいいけれど，それだと時として不便だ。私は派手に一元管理がしたい。
そんなとき，データベースソフトが必要になる。CARD PROはそういったものだ。会社で使うような場合は置いておいて，個人で使うにしても，大人だから，いろいろと余計な知り合いとかも増えるわけだ。
私が好んでいるのは，個人か法人かの分類項目。だって，人名の間に“Oh!X編集部”なんてあると，変。それでもって，男

か女かとか，互いの知り合った基盤とか(高校時代の同級生とか，会社の同僚とか)，そんな項目があると便利だ。検索も速い。

CARD PROはマウスを使って参照画面のレイアウトができたり印刷画面のレイアウトができたりと，住所録おたくには欠かせない機能を満載している。

ただ，項目の設計が面倒だ。某国民機や世界でいちばん売れているマシンになると，項目の長さを指定する必要がない（不定長項目！）ものや，一覧表画面でデータ入力できるものもある。CARD PROだけでなく，そういった新しいタイプの柔軟データベースも欲しいところである。

挫折しないための住所録

人間，年を取ると閑に任せてデータをドタドタと打ち込んでいったりする。しかし，そうなるまで年を取るのはなかなか億劫である。

そこで，私はステップバイステップ方式を提唱したいと思う。すでに1000件以上データの詰まったCARD PROなんかを使っている人は読み飛ばしていいよ，と。

エディタ&ワープロ→Kamikaze（あるいはStationeryかCYBERNOTE）→CARD PRO。

ステップ1

エディタで住所録を作る。とりあえず，まだ見ぬ明日を考えて，項目ごとの区切りを統一させるのが賢明だ。使い込んでいくうちに現状の環境に満足できなくなったとき，変なデータ管理をしていると，変換プログラムを自分で書いたり，面倒な手作業を必要としたり，最悪の場合，再度打ち込み直すことになりかねない。で，どういうものが考えられるかというと，

1)　タブ

2)　カンマやら@など，データ中になくて紛らわしくない文字

1) &2) の方法がいちばんおいしい。

だが，これだと，タブ→スペース変換プログラムが必要となる。タブを使うメリットは，桁合わせが楽なことと日本語入力モードでもシュタっとカーソルを飛ばせることに尽きる。大人は楽をしたがるものである。

で，X68000ユーザーはみんな，タブ→スペース変換プログラムを持っている。なんのことはない，付属ワープロのことだ。あのワープロは勝手にタブ→スペース変換をしてくれるのだ（良くも悪くも）。

例を図1に示す。区切りには“/”を使った。なんとなく，それらしいから。

ここで普通の大人は思うわけだ。エディタじゃあ，印刷に困るじゃん。んでもって，印刷プロセッサとしてワープロが働くのである。エディタ&ワープロというのはコストの点で秀逸だ（ただだもん）。

まず，元の住所録をコピーして，ワークファイルを作る。そいつをエディタに呼び出し，置換機能を利用して/をスペースに変える。そんでもって，印刷に必要な項目以外は削除する。

桁合わせしてある限り同じ作業の繰り返しになるので，ED.Xのキーボードマクロを使うと簡単だ。

図1の例から出身地を消してみる。なんでこんな迷惑な場所に出身地があったかというと，ここで消してしまうためだったりするのだ。うーむ，作為的。

まず，データの先頭の行の頭にカーソルを置いて，それからだ。

1)　ESC−@でマクロ登録モードに入る。

2)　CTRL+Fで，1語右へ移動して出身地の項目へカーソルを飛ばす。

3)　CTRL+Gで，/を削除する。

4)　CTRL+Tで，1語削除する。

5)　CTRL+Xで，1行下へカーソルを移動し，

6)　CTRL+Qで次の行の頭へ行く。

ここで，UNDOキーを押せば，よきにはからっててれてれと全部消してくれるはず，である。終わったら，ESCキーでマクロ実行モードから抜けるべし。

しかし世の中ままならぬ。うまくいかないケースなんて山ほどあるのだ。

たとえば，エディタにとっての1語というのが何かを知っておく必要がある。普通に考えて，名字と名前と出身地で3語のように見えるが，そこはそれ，見えないところに秘密があるのである。名字と名前の間には“全角のスペース”が，名前と出身地の間には“半角のスペース”（あるいはタブコード）が埋まっているのだ。半角のスペースやタブコード，カンマ，ピリオド，ハイフンなどなどアメリカンな感覚でセパレータなものが1ワードの区切りなのであって，ジャパニーズな全角文字はその限りではないのである。ED.Xは昔ながらのイングリッシュな仕様なのであった。

そんなこんなで，必要な項目だけにシェイプアップしたデータを，ワープロのテキスト入力で読み込む。あとはワープロさんにお任せで，罫線引くなりなんなりすればいい。

ステップ2

どうしてエディタを使ったかというと，少ないデータなら，すぐに呼び出せたほうが便利だからである。項目数が少ないなら，エディタのほうが追加・更新が楽だからである。しかし，データ数が増えたりすると，手作りの味にも限界が見えてくる。

私は個人的な好みから，ここでKamikazeに移る。

図1のエディタで作ったデータを，Kamikazeで扱えるCSV形式にコンバートしてやるわけだな。

図1では/を項目の区切りとして採用した。それを，「"」で括って，「,」で分ける形式にするわけである。

まず，行頭と行末は強制的に"であるから，上の要領でマクロを作り，"を挿入してしまおう。

続いて，置換する。「/」を「“,”」に置換してやればうまくいくだろうというのは鶏頭でもわかることだ。

エディタの置換機能で，全置換する。というわけで，CSV形式ファイルの出来上がりである。項目間がタブで区切ってあれば，置換機能でタブを“ ”（ヌル）に置換する。“前方置換旧文字列”をタブにし(HTと表示される)，“前方置換新文字列”のところでただリターンを押せばいいのだ。めでたしめでたし（図3）。

これを適当な名前でセーブして，Kamikazeで呼び出す。

Kamikazeの表計算シートを開いて，ファイルを読み込む。Kamikazeはどんなファイルか勝手に判断して読み込んでくれるので，そのまま読み込みをすればいい。たいていうまくいく（図4）。

ステップ3

なにもKamikazeに読み込む必要はない。目的によってCYBERNOTEでもCARD PROでもいい。

で，CARD PROに読み込むことを考えてみよう，と。これにはCARD PROを立ち上げて“ファイル”のメニューから“コンバート”を選ぶ。そんでもって，形式を選ぶ。転送元ファイルと転送先ファイル（こっちはデータベース名）を指定する。

この形式だが，まあ，CSVか区切りなしASCIIになるだろう。CSVの場合はステップ2と同じようなものなのでよい。区切りなしASCIIの話をする。

CARD PROでいう区切りなしASCIIというのは先月説明したとおりなので，図1から置換機能で“/”を取り，余計な行を消して，タブをスペースに置き換えたものを使う。タブ→スペースはファイルをワープロでファイル入力して，ファイル出力するだけである。

このコンバートをする際には2つの手作業が必要になる。ひとつは，“CARD PROで区切りなしASCIIファイルを読むときは，各レコード（この場合は1行1レコード）の先頭に4文字の半角スペースが必要なのだ!”ということ。初めからCARD PROを念頭においているなら4文字開けておくのもいいが，そうでないなら，エディタのマクロなどを使って，なるべく楽をして4文字分の空白を作っておこう。

2つ目は，数えることである。区切りがないということは，項目間の区切りはユーザーの頭の中にだけ存在するというわけで，最初の項目が何桁で（もちろん半角で計算ね），2番目が何桁でっていうのをメモしておかねばならない。コンバート時に項目はいくつでそれぞれの桁数を入力してやる必要があるからだ。なお，このとき，タブコードが混じっているとCARD PROが桁数計算を間違えてくれたりするので気をつけるように。

無事終了すると（結構早い），ついさっきまでエディタで作っていたテキストファイルがちゃんとしたデータベースになっていたりしてなかなかパソコンの醍醐味を味わわせてくれる（図5）。項目の桁数を数え間違えるとなかなか笑える結果になるので，うまくいかないときはまず自分を疑ってみるのが得策だ。

＊

PC-9801上で動くアプリケーションとのデータ互換性はどうなってるんだ！　という人も多いと思うので，そんな話題もおいおい取り上げていくかもしれない。

いーかげんに始めたわりには期待する人が多くてビビっている荻窪“のーてんき”圭は，Kamikaze入門なるものを考えているのであった。では来月。

図1

TABと“/”で区別した電話番号簿のED.X
‘→‥‥’はTABを
（↓）はCR/LFを
[EOF]はファイルの終わりを表示させてある。

```
氏名‥‥‥‥出身地‥電話1‥‥‥‥電話2(勤務先)‥勤務先‥電話3(いそうなところ)‥3の場所↓
代々木　圭‥‥/鳥取県‥/03-263-****‥‥/‥‥‥‥‥‥/浪人‥/03-963-****‥‥/↓
新宿　圭‥‥‥/東京都‥/03-239-****‥‥/‥‥‥‥‥‥/なし‥/‥‥‥‥‥‥/↓
大久保　圭‥‥/千葉県‥/03-328-****‥‥/03-339-****‥/Ｙの家‥/‥‥‥‥‥‥/↓
東中野　圭‥‥/埼玉県‥/03-352-****‥‥/03-638-****‥/公務員‥/‥‥‥‥‥‥/↓
中野　圭‥‥‥/東京都‥/03-368-****‥‥/03-***-0101‥/○井‥/‥‥‥‥‥‥/↓
高円寺　圭‥‥/宮城県‥/03-373-****‥‥/‥‥‥‥‥‥/なし‥/‥‥‥‥‥‥/↓
阿佐谷　圭‥‥/愛知県‥/03-392-****‥‥/03-793-****‥/M銀行‥/‥‥‥‥‥‥/↓
荻窪　圭‥‥‥/愛知県‥/03-395-****‥‥/‥‥‥‥‥‥/なし‥/03-5488-1309‥/Oh!X編集部↓
西荻窪　圭‥‥/東京都‥/03-385-****‥‥/‥‥‥‥‥‥/なし‥/‥‥‥‥‥‥/↓
吉祥寺　圭‥‥/長野県‥/0422-55-****‥/03-266-****‥/C広告‥/03-725-****‥/女の部屋↓
三鷹　圭‥‥‥/兵庫県‥/0422-23-****‥/‥‥‥‥‥‥/学生‥/‥‥‥‥‥‥/↓
[EOF]

14 行　空領域　952644　挿入
```

図2

図1からED.Xのキーボードマクロを使って，出身地を消したもの(データのみ)。

```
代々木　圭　　/03-263-****　　/　　　　　　　/浪人　/03-963-****　　/
新宿　圭　　　/03-239-****　　/　　　　　　　/なし　/　　　　　　　/
大久保　圭　　/03-328-****　　/03-339-****　/Ｙの家 /　　　　　　　/
東中野　圭　　/03-352-****　　/03-638-****　/公務員 /　　　　　　　/
中野　圭　　　/03-368-****　　/03-***-0101　/○井　/　　　　　　　/
高円寺　圭　　/03-373-****　　/　　　　　　　/なし　/　　　　　　　/
阿佐谷　圭　　/03-392-****　　/03-793-****　/M銀行 /　　　　　　　/
荻窪　圭　　　/03-395-****　　/　　　　　　　/なし　/03-5488-1309　/Oh!X編集部
西荻窪　圭　　/03-385-****　　/　　　　　　　/なし　/　　　　　　　/
吉祥寺　圭　　/0422-55-****　 /03-266-****　/C広告 /03-725-****　　/女の部屋
三鷹　圭　　　/0422-23-****　 /　　　　　　　/学生　/　　　　　　　/
```

図3

図1をCSV形式に変換したもの。作業は1分とかからず，簡単に終了。

```
"代々木　圭","鳥取県","03-263-****","","浪人","03-963-****",""
"新宿　圭","東京都","03-239-****","","なし","",""
"大久保　圭","千葉県","03-328-****","03-339-****","Ｙの家","",""
"東中野　圭","埼玉県","03-352-****","03-638-****","公務員","",""
"中野　圭","東京都","03-368-****","03-***-0101","○井","",""
"高円寺　圭","宮城県","03-373-****","","なし","",""
"阿佐谷　圭","愛知県","03-392-****","03-793-****","M銀行","",""
"荻窪　圭","愛知県","03-395-****","","なし","03-5488-1309","Oh!X編集部"
"西荻窪　圭","東京都","03-385-****","","なし","",""
"吉祥寺　圭","長野県","0422-55-****","03-266-****","C広告","03-725-****","女の部屋"
"三鷹　圭","兵庫県","0422-23-****","","学生","",""
```

図4

図3をKamikazeに読み込んだもの。項目長(セル幅)の調節はしていない。

図5

CARD PRO-68Kで読み込んだものを一覧表表示してみた。項目名はまだつけていない。

特別付録　データの移動・関連図

ハードウェア工作入門《4》

A/Dコンバータ その1

Misawa Kazuhiko
三沢 和彦

基本インタフェイスが一段落し，今月からいよいよ第2部に入ります。テーマは実用性も高いA/Dコンバータを取り上げましょう。とても簡単な工作ですが，これをひとつ作っておくとあとあと楽しみが広がりますよ。

第1部の基本インタフェイス回路はいかがでしたか？　初めての教材ということで，回路の単純さと工作のしやすさに重点を置いたため，回路自体は実用性のないものになってしまいました。しかし，外部機器のコントロールはそれほど難しくないということがわかってもらえたことと思います。

さて，第2部は，回路が簡単なうえに，実用性もあり，さらにさまざまな応用も利くという極めつけの回路にチャレンジしてみることにします。

A/Dコンバータとは何か

A/Dコンバータとは，文字どおり，アナログ量（A）をデジタル量（D）に変換（コンバート）するものです。そもそも，アナログ量とデジタル量の違いは，

アナログ＝連続的に変化する量

デジタル＝離散的(階段状)に変化する量

と定義されていることが多いようです。コンピュータで扱う量は，すべてデジタル量なのですが，自然界に起こっていることはほぼすべてアナログ量といってよいでしょう。人間の五官が感じる量も，光の明暗，音の強弱，温度の高低など，連続的に変化しているアナログ量ばかりです。

もちろん，これら自然の量について人間が感じるときには，たとえば夏は暑いが冬は寒いなどと，だいたいのことをいえばすんでしまいます。ところが，正確に測定し記録するときは，夏の平均気温は30℃で高いが，冬は10℃で低いというように数字で表す必要がありますね。この数字で表すというのが問題で，たとえば30℃というのが30.2℃なのか30.3℃なのか，さらには30.26℃なのか30.27℃なのか細かいことをいいだすとキリがありません。

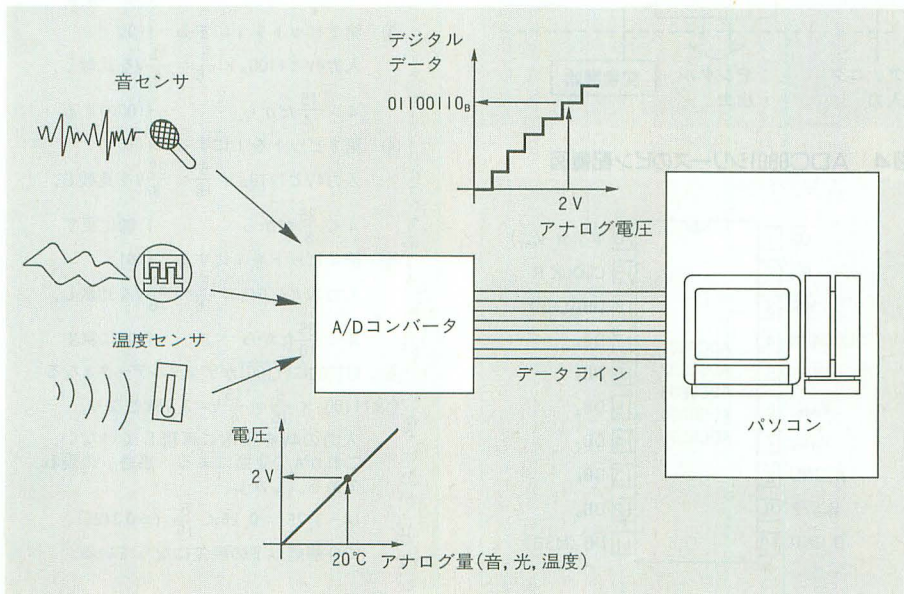
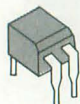
これらを有限の時間で処理するには，どこかの桁で打ち切る必要があります。いわんや，コンピュータに処理させる際にはだいたいいくらではまったく許されず（近年技術の進んできたファジィコンピュータはこの欠点も克服しています），すべてキリのいいデジタル量で表現しなければなりません。

一方，身近な家庭において，機械が人間の感覚の代わりに自然界の量を計測するシステムが普及しています。たとえば，エアコンが現在の室温を検知して自動的に温度を調節するシステムでは，まさに人間の代わりに暑い寒いをコンピュータが「感じて」いるわけです。この温度調節のシステムのほかにも，テレビのリモコンでは光を感じることによってスイッチのON/OFFをコントロールしたり，火災報知器では煙やガスを感じることによって警報を鳴らしたり，自然界のアナログ量を計測するシステムが非常に多く使われています。

こうしたシステムの内部では，アナログ量はすべてデジタル量に変換されてコンピュータが処理しています。また，皆さんにはもっとも身近と思われるコンパクトディスク（CD）でも音というアナログ量をデジタルデータに変換したのちに，ディスクに1または0のデジタルデータの列を刻み込んでいるのです。さらには，X68000にも載っている音声サンプリング回路も代表的なA/D変換回路の例です。

このようなA/D変換を行うシステムにおいては，まずシステムに入力されるアナログ量がなんらかの形で電圧に変換されます（この仕組みは第3部に予定しているセンサ回路のところで，非常に詳しく解説するつもりです)。そして，A/Dコンバータは入力されたアナログ電圧を8（〜16）ビットのデジタルデータとして出力するのです。したがって，本当の意味でのA/D変換回路は，自然のアナログ量を電圧に変えるセンサ部とそのアナログ電圧をデジタルデータに変換するコンバータ部の2つから成っています。そして，データを変換したのちにそのデータをパソコンに受け渡しするという意味では，A/Dコンバータもインタフェイスのひとつと考えてもよいでしょう。

今回は入力電圧をデジタルデータに変換する部分に絞って説明していくことにします。

図1　A/D変換システム

A/Dコンバータの機能と仕組み

A/Dコンバータには，

1) 逐次比較型
2) 積分型
3) 並列比較型
4) V-Fコンバータ

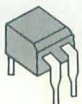
など，用途によっていろいろな方式がありますが，どの方式でも基準電圧V_{ref}を設定し，その基準電圧と測定すべき入力電圧とを比較しながら，デジタルデータに変換します。ここでは，もっともよく使われる逐次比較型について具体的に説明しましょう。今回の回路に使うのもこの方式です。

ここに8ビット出力，基準電圧V_{ref}のA/Dコンバータがあるとします。逐次比較型は，8ビットのデジタルデータを記憶しておく「逐次比較レジスタ」，そのデジタルデータを実際の電圧値に変える「D/Aコンバータ」，D/Aコンバータからの電圧を外部入力電圧と比較する「コンパレータ」とから成っています（図2）。

このとき，D/Aコンバータは8ビットの逐次比較レジスタの値によって256段階の電圧を出力します。具体的にいうと，逐次比較レジスタの上から第nビット（最上位ビットを第1ビット，最下位ビットを第8ビットとします）に1が立っているとき，基準電圧V_{ref}に対し，$V_{ref}/2^n$を足し込んでいきます。たとえば，逐次比較レジスタが10101101のときは，$V_{ref} \times (1/2+1/8+1/32+1/64+1/256) = V_{ref} \times 173/256$がD/Aコンバータの出力です。

すなわち，逐次比較レジスタの値を2進数としたとき（$10101101_n=173$），その値に$V_{ref}/256$を掛けたものが電圧値になっているわけです。したがって，逐次比較レジスタによって，$V_{ref}/256$の分解能で0～V_{ref}の電圧値を256段階で表すことができるわけです。これが，デジタルデータでアナログ電圧を表現する仕組みです。

では，変換の仕組みに説明を移しましょう。まず，最初に逐次比較レジスタをリセットして0にします。変換開始とともに，逐次比較レジスタの最上位ビットを1にします。ここで，D/Aコンバータの出力は$V_{ref}/2$になっています。コンパレータで入力電圧と比較し，入力電圧のほうが大きければそのままビットに1を立てます。次に第2ビットに1を立て，D/Aコンバータの出力$V_{ref} \times (1/2+1/4)$と入力電圧との大小を比較します。こうして，各ビットごとに「逐次比較」していって，入力電圧にもっとも近い値を示すデジタルデータを見つけます（図3参照）。

実際に回路に使うA/Dコンバータは，上で述べた機能が1個のICの中に入って製品化されていますから，ユーザーはICを買ってくるだけで手軽に使うことができます。また，A/Dコンバータは使い道が広く，たいへん普及しているので，専用ICも非常に安価に入手できます。

パソコンとのインタフェイス

ほとんどすべてのA/Dコンバータ用ICはコンピュータと接続することを前提に作られているため，コンピュータのI/Oポートにそのままつなぐことのできるものばかりです。図4は非常によく使われる8ビットA/DコンバータのADC0801シリーズの端子を示したものです。見てのとおり，デジタルデータの出力はDB_0～DB_7までの8ビットバスラインに直接出てきますので，これをそのままI/Oポートのバスラインにつなげばよいのです。

ところが，私たちのこの連載ではパソコンとのインタフェイスはジョイスティックポートに限ることにしています。前回までの記事でわかるとおり，ジョイスティックポートの入力は4ビットしかありません。したがって，普通のICはそのままつなげないことになります。多くの工作記事では「入門編」と称しながら，ここでいきなりジョイスティックポートを捨てて，拡張ボードに作り変えるという掟破りの技に出るところでしょう。しかしご安心ください。ちょっとした工夫で，8ビットA/Dコンバータをジョイスティックポートにつなぐことができるのです。

その工夫はシリアルインタフェイスにあります。シリアルインタフェイスは1本の

図2 逐次比較型A/Dコンバータのブロック図

図4 ADC0801シリーズのピン配置図

図3 逐次変換の手順

5Vフルスケール（V_{ref}=5V）で4ビットのデジタルデータに交換するとき，

$\frac{V_{ref}}{2^4}=\frac{5}{16}$Vが分解能になる。

◎例として4Vを入力してみる

① オールリセット 0000

② 第1ビットを1にする 1000

入力4Vと$1000_B \times \frac{5}{16}=\frac{15}{2}$Vを比較し，

$4>\frac{5}{2}$だから 1000のまま

③ 第2ビットを1にする 1100

入力4Vと$1100_B \times \frac{5}{16}=\frac{15}{4}$Vを比較し，

$4>\frac{15}{4}$だから 1100のまま

④ 第3ビットを1にする 1110

入力4Vと$1110_B \times \frac{5}{16}=\frac{35}{8}$Vを比較し，

$4<\frac{35}{8}$だから 1100に戻す

⑤ 第4ビットを1にする 1101

入力4Vと$1101_B \times \frac{5}{16}=\frac{65}{16}$Vを比較し，

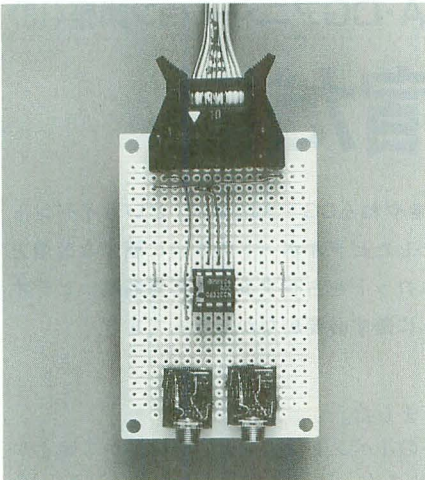
$4<\frac{65}{16}$だから 1100に戻す

⑥ 最終的に1100がデジタルデータとなる

（注）$1100_B \times \frac{5}{16}=\frac{15}{4}$V=3.75Vとなり，

入力の4Vを正確に再現していない。これがA/D変換による「誤差」の現れである。しかし，

$14-3.751=0.25<\frac{5}{16}$（=0.3125）

で分解能以下の誤差になっている。

信号線（実際はGNDラインと合わせて2本）だけで8ビットデータを順次送る方式で，8本の信号線で8ビットまとめてデータを送るパラレルインタフェイスと比べ，信号線が少なくてすみます（図5）。反面，1回のデータ転送にかかる時間はパラレルインタフェイスよりも多くなります。身近なシリアルインタフェイスとしてRS-232CやMIDIなどがあるのはご存じでしょう。

この方法を使えば，信号線の本数に制限のあるジョイスティックポートでも8ビットデータの転送が可能となるのです。このシリアルインタフェイスは今後の記事でも使おうと思っていますので，ここで，シリアルインタフェイスの仕組みをごく簡単に解説しておきます。

シリアルインタフェイスでもっとも問題になるのは，何ビット目のデータがいつ来るかのタイミングをとらなければならないことです。まず最初に考えられるのはあらかじめ決められた時間間隔で規則正しくデータを送り，受ける側も同じ時間間隔で区切ってデータを取り込むという方法です。この決められた時間間隔のことをボーレートといいます。

このとき，8ビットデータの初めと終わりを知らせるために，スタートビットとストップビットとを付けています。この時間間隔を決めるクロックを送受信側別々に持てば，送信側はデータを一方的に送り続け，受信側はひたすら受け続けることになります。ただし，この方法では1回でもタイミングが狂うとそれ以降のビットがすべてずれてしまい，データが無意味なものになってしまう恐れがあります。そこで，信号線をもう1本増やして送信側か受信側かのどちらかからクロックを送り，送受信で完全に同期をとってデータを転送すれば，信頼性が向上します。

以上がシリアルインタフェイスのデータ転送の一般論ですが，具体的なデータ転送のやり方は今回の回路を実際に見ながら説明していこうと思います。

A/Dコンバータ回路の実際

図6が今回の回路です。前置きが長かったわりには，騙されているのかと思うくらい簡単な回路でしょう。それは，超高性能なA/Dコンバータ専用ICのADC0832を選んだおかげです。

今回の回路は主要な部品がこのICしかないので，回路の説明というよりは，このICの機能の説明ということになります。このICだけで，これまで述べた逐次比較型A/Dコンバータ＋シリアルインタフェイスの機能を2系統持っています。しかも足は8本だけで，工作は極めて簡単です。

まず各ピンの役割をざっと説明しておきましょう。

8番の＋5Vと4番のGNDは問題なし。2番と3番がアナログ電圧の入力端子で，どちらもまったく同じ機能で，それぞれ0～＋5Vの入力が許されています。また万一この範囲を瞬間的にオーバーしたときのための簡単な保護回路も内蔵されています。そして，この2つ（CH0，CH1）のどちらの入力をA/D変換するかはパソコン側から制御できるようになっています。

1番ピンは$\overline{\mathrm{CS}}$(チップセレクト)といって，この端子をLにしたとき，このICが機能します。通常はHにしておいて，A/D変換を始めるときにLにしてやるとICがA/D変換をスタートします。パソコン側からコントロールしてやることによって，変換されたデジタルデータの最初のビットがどこから始まるかをパソコンが知ることができるのです。

7番ピンが，先ほど説明したようにシリアルインタフェイスのためのクロックをICに入力する端子です。実は，このクロックは一定間隔でなくてもよいのです（来月詳しく説明します）。

5番ピンのDI（データイン）はこのICの使用モードを設定するためのデータをパソコンから送ってやる端子で，このデータもやはりクロックにしたがってシリアル転送します。この端子の使い方も来月詳しく説明します。

最後の6番ピンのDO（データアウト）は文字どおり，A/D変換されたデジタルデータがシリアルで出てくる端子です。

といったところで，これらの端子を直接ジョイスティックポートにつなげば終わりという，とんでもなく簡単な工作になります。

このADC0832を使うと，基本的にはこれ以上部品は必要ありません。実は，もう少し外付け部品を加えて入力の電圧範囲を自由に設定できたりするような回路も一応試作してみました。しかし，入門編ということなので最小限の回路で最大限楽しめるようにと，思い切っておまけの部分を切り捨てたしだいです。

*

今月はA/Dコンバータの一般論に重点を置いて説明しました。回路自体はなにも考えることがないくらい単純ですが，それだけに今回用いた専用IC（ADC0832）の使い方（プログラミング）をしっかりマスターしておかなくてはなりません。次回は工作編ですが，工作自体は極めて簡単なため，工作手順については簡単に説明をすませ，ADC0832の使い方をじっくり研究することにします。X-BASICによるドライバプログラムも載せる予定です。それでは，お楽しみに。

図6　回路図

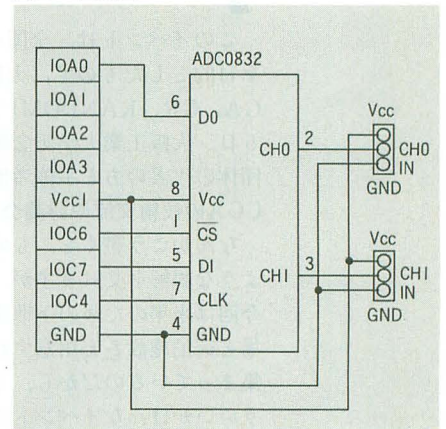
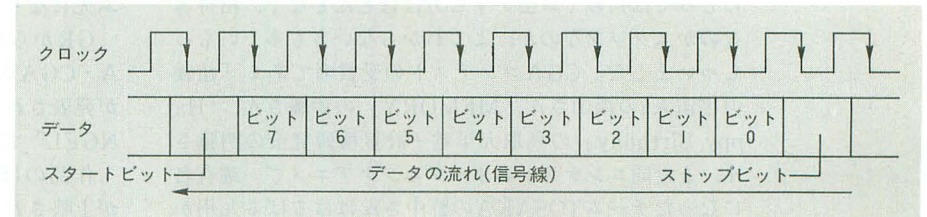



図5　シリアルインタフェイスのデータ転送

CGA緊急レポート 夏だ,祭だ,合宿だ!

プロジェクトチーム DōGA かまた ゆたか

先月号の予告どおり「好きやねんCG 大阪合宿」というイベントの紹介と,先頃発売されましたビデオボードの本当に簡単な改造方法をレポートします。2スロットを占有することで悪名高いビデオボードは,本当はスロットに差す必要がなかったんです!?

はじめに

もうすぐ「花博」も終わるようですが，今年の夏は「花博」のついでに当プロジェクトルームに遊びに来る方がたくさんいました。一般ユーザーだけではなく，某大手メーカーやソフトハウスの方もいらっしゃいました。なんの用があるんだろうと思いつつも相手をしていると，“あっそろそろ新幹線の時間ですので”といって帰ってしまう（新大阪駅にわりと近い)。ここは新幹線の待合室ではありませんよ。

しかし，げに恐ろしきは女子高生なり。千葉県のTさん。この方は以前にもいらしたことがあるとはいうものの，うちのクラブ（大阪大学コンピュータクラブ）の夏合宿に，いつの間にか参加していました。それから名古屋のYさん，アポイントなしに寝込みを襲うのはやめてください。朝8時に起きているスタッフなんていません。皆さんも遊びに来るときは，事前に日時などご連絡ください。

好きやねんCG 大阪合宿

このイベントは，全国のアマチュアCGA団体の交流を目的としたもので，大阪で活動している3団体（DōGA，GR，RANDOM）で共同主催しました。8月4，5日，大阪工業大学記念館にて，全国から招待された各団体の代表の方と主催者側のスタッフが一緒になって，CGAの技術交流や討論会などが中心となっています。

なんかこう書くと，ものすごく立派なイベントだったような誤解を受けますが，それは来年からということで，今回は来年のための下準備といった程度のものです。会場と宿泊施設さえ用意すれば，あとは同じ穴のムジナが集まっているのだから，それなりに楽しめるだろうといういいかげんなイベントでした。

参加者は招待客が約20名，スタッフが約20名です。私にとっては初めてお会いする方はほとんどなく，招待客なのかスタッフなのか，よくわからない方も多くいらっしゃいました。CGAコンテストの受賞者である「超強力宇宙人」の森山さん，「MEMORY」の伊藤さん，「Happy Birthday」の鳥取大学電子計算機研究会の門脇さん，また同コンテストのオープニングアニメで一躍有名になったチームTOSAKAの野中さんははるばる九州からいらっしゃいました。

まず，この手のイベントは主催者側の挨拶から始まります。GRの森山さん，DōGAを代表して私，RANDOMを代表して小味さんです。GRの森山さんは，本誌の6月号でレンダリングソフト「ANGEL」を発表されたので皆さんもご存じでしょう。GR（グラフィックス研究会）は，大阪工業大学のサークルで，今回のイベントの中心となって準備してくださいました。RANDOMは大阪府立大学のコンピュータクラブで，小味さんは昨年のCGAコンテストでも「Let me Dance!」で奨励賞を受賞しています。さて，この3人の挨拶ですが，ご多分にもれず，この手の挨拶には内容がないと相場が決まっておりますので，割愛させていただきます。

形式的な自己紹介のあと，実際にマシンを触りながらの勉強会です。まずは，DōGAからオリジナルウィンドウシステム“Ko-WINDOW”が発表されました。これは，prodige（大阪大学コンピュータクラブ）の小林君が昨年から趣味でひそかに構築していたもので，一口で言うとDōGA版のSX-WINDOWです。どちらが優れているかなんてことは私にはよくわかりませんが，ただひとつはっきり優れていると言えるのは，仕様が公開されていて，わからないことがあれば，開発者本人に聞けるという点です。ということで，今DōGAではこの“Ko-WINDOW”上のプログラムを開発するのが流行っています。

さらに，KMC（京大マイコンクラブ）の石井さんや柘植さんから最近KMCで開発された（ている）プログラムが発表されました。なかでも目を引いたのが，TOWNS版やPC-9801版のアニメーションプログラムです。TOWNS版はX68000版より高速ですし，PC-9801版は色数が足りないのを，タイリングやモノクロ16諧調で表示することでカバーしていました。さらに，X68000版の512×512のアニメーションがついに発表され，一同を驚かせました。しかし，これらのプログラムはまだ問題点があり，一般に配布されるようになるのは，まだだいぶ先になりそうです。

GRからは，“ANGEL”の最新バージョンと，DōGA・CGAシステムとのデータコンバータ（形状データ）が発表されました。これで，CADで作ったデータを“ANGEL”で作画することができます。

作品のほうでは，DōGAで制作された実験的な映像が上映されたり，CG連合の古い作品が公開されました。

また，DōGAの砂川君が数カ月をかけてデザインしたZガ○ダムの変形シーンや，面数の多さでは日本一のチームTOSAKAのグリフォ○などが職人の意地を見せてくれました。私は，形状データは少ない面数でいかに表現するかが大切だと思っていますが，野中さんは“面数こそ命！”と言いきっていました（グリフォ○のデータは800Kバイトあります）。

なんかこう書くと，とっても有意義なイベントだったみたいですが，実際はマシンを囲んで雑談しているだけで，内容もなくだらけたものになっていました。詳しいことは割愛させていただきます。

夜に入って，いよいよ宴会が始まります。ここら辺でちょっとずつ盛り上がってくる……と思いきや酒や肴も予算の都合で足りず，各自が雑談しているだけという昼間の延長のようになってしまいました。もちろん，詳しいことは割愛させていただきます。

さて，10時を過ぎると１日目の予定は終了し，会場も閉められ，各自の部屋に戻って就寝の時間です。このままでは明日も思いやられるなぁ，と暗い気持ちで皆さん眠りにつき……なんてことはありません。

さぁ！　いよいよメインイベントです。今までのはただの形式，お互いの顔と名前を覚える儀式にすぎません。名づけて“今夜も朝までパワフルCGA討論会！”。

朝まで討論会

この討論会のすべてはカセットテープに録音されています。あとでワープロで打ち直すと，雑談や聞き取れないところを省略しても80ページという膨大な量になりました。CGAシステムの問題点，アマチュアのネットワークのあり方，実にさまざまな論議が非常に活発に行われました（おかげさまで２日目の午前中は，皆さん寝ぼけて何もできないありさまです）。もちろん，そのすべてを掲載するわけにはいきませんが，ほんのさわりだけでも紹介しましょう。

*

——アマチュアの横のつながりをどうしていくかという問題について，ご意見や現在の活動の問題点などありましたら。

▶「こういう合宿を年１回やってください（笑）」

▶「それはありますよね。現時点では，CGに興味をもっているすべての一般の人が実際にCGA制作をする段階ではなく，DōGAにCGAシステムを申し込むような積極的な，マニアックな人たちが先行して活動する時期だと思う」

▶「問題は作品で，作品を通じてのコミュニケーションがいちばん大切じゃないですか。そして，私たちは上映会とか，発表する場を提供していかなくては」

——そういうことで，CGAコンテストを始めたんですけど（笑）。

▶「SF大会でもやりますんでよろしく（笑）」

▶「上映会とかコンテストはたくさんあったほうがよいです。そのわりに作品がないのは問題ですけどね」

▶「自主制作映画界ってのが衰退気味というのがあるでしょ。実写とかアニメを含めたもっと大きい世界で」

——もしそうだとしたら，これだけ映像が普及している世の中でなぜ映像を作ることが衰退しているんだろう。

▶「それは簡単な話で，作る必要がないからです」

▶「いや，それを言うんなら，バンドだって，バイクだって，X68000だって必要ない（笑）。必要がないのに流行っていることなんか山ほどある。趣味なんてそんなものだ」

▶「今の学校教育では，自己表現する必要性がない。その欲求すら失いつつある人のほうが多い」

▶「もしそうなら，とても悲しいが，すべての人がそうだとは思わない」

▶「機材の問題が大きいのじゃないかな。音楽は口だけでもできるけど，映像は機材とかいるし，作る段階までいく人間が少ないでしょう」

——確かに初心者がとっつきにくいという問題は大きいと思う。では，そのとっつきにくさが，みんなの努力によって改善されていけば，どんどん映像を作る人たちが増えてくるでしょうか？　もしそうなら，私達は悩む必要もなく，CGAシステムのバージョンアップなど，今までの活動を続けていけばいい。

▶「それでいいと思いますよ。今はたまたま８㎜が衰退

CGAコンテスト事務局より

さて，いよいよ第３回アマチュアCGAコンテストの募集が始まります。DōGAプロジェクトルーム内にコンテスト事務局も設置されました。今回は，賞の形態や審査員を一新し，より本格的なコンテストを目指して，ちゃくちゃくと準備が進められています。応募要項など詳しいことは来月号にて発表されますが，とにかく今年の12月31日が締め切りであることは間違いありません。皆さんもラストスパートをお願いいたします。

“朝まで討論会”でもあったように，アマチュアCGAの普及には質のよい，たくさんのCGA作品が不可欠であり，皆さんにとっても作品の制作と発表は個人レベルの活動の成果であり，当プロジェクトへの参加の道といえるでしょう。当チームが主催するコンテストである以上，当チームから作品を出品するわけにはいきません。このコンテストの成功は，読者の皆さんにかかっているのです。

このコンテストは，入賞することより入選することに意義があります。それは，このコンテストの主旨が，より多くの方々の作品の発表の場であるからです。とても賞なんてとれそうにないなんて言わないで，積極的に応募してください。また，“今年は間に合いそうもないから来年には”などといっていては，いつまでたっても決して完成しません。今からでも，ひとつの作品を完成することは十分可能です。努力と根性でぜひがんばってください。たくさんのご応募をお待ちしております。

しちゃって作る手段がなくなっているから，落ち込んでいるのであって，手段ができれば復活するでしょ」
▶「ぼくはあまりそうだとは思わない。作る手段としてはビデオカメラがちゃんとある。8㎜よりビデオのほうが楽でしょ。すぐその場で見られるんだから」
——ビデオカメラがこれだけ売れているのに，自主映像制作は復活の兆しがない？
▶「映写施設の問題でしょ。いちばん最初は，8㎜でもビデオでも適当に撮ったものを自宅に持ち帰って，友人2，3人で笑いながら見る。そして，これは受けるぞと思ったら，もっとアイデアを温めて，大がかりに制作して，上映会というメディアに参加していく。みんなそうやって自主制作の世界に入ってきた。しかし，ビデオの場合，一度にたくさんの人が作品を見ることができない」
▶「その問題は大きいと思う。CGAにしろ，各個人の活動の基本は，作品を作り，見せ合うということだから。見せ合う場がなければ，作品制作意欲も出ない」
▶「また，多くの観客に切磋琢磨される機会がなければ，質のよい作品は生まれない」
▶「コンテストや，上映会に参加することで，お互いに触発されますからね」
——でも，最近液晶プロジェクタが出てきましたよね？あれなら上映会に手ごろなんじゃないですか。
▶「50万円ですよね。あんなの持っている人がいますか？」
▶「70インチで10万円ぐらいのが，こないだ出たよ」
▶「そりゃ凄い。解像度はいいんですか？」
▶「8㎜より悪いことはないと思いますよ（笑）」
▶「画質とかは根本的な問題ではないと思う。それに，価格や画質の問題はこれからどんどん改善されるでしょう」
▶「それなら，自主映像制作の未来は明るい」
▶「しかし，ビデオは父親が子供を写す道具というイメージ，つまりかっこ悪いというイメージが定着しちゃって，若い世代が使いたがらないという話を聞くが」
▶「それはかっこいい作品を見せることで，感化すればいい」
——とすると，あとは教育の問題と機材，とっつきやすさの問題だけですね。でも，教育の問題は我々が解決すべきことではないでしょう。
▶「そう，CGA特有の問題を検討すればいい。でも，機材とかとっつきやすさとかも，プロジェクタと同様どんどんよくなるに決まってますよ」
▶「ただ，ちょっと心配なのは，使いにくいとはいえ，CGAシステムはX68000ユーザーにかなり広まっていて，バージョンアップサービスを希望する，つまりそれなりに使っている人がたくさんいるのに，作品が集まっていないという点です」
▶「CGAシステムを一生懸命開発して，苦労して配布したのは，作品を作ってもらうためで，本棚に飾ってもらうためじゃない」
▶「でも実際問題として，CGA作品をひとりで作るというのはとても難しいと思う。やっぱり仲間がほしくなる」
▶「そうかな。作品っていうのは，基本的にひとりで作るもんだと思うけど」
▶「そうじゃなくて，作品制作はひとりで行うんだけど，チームの中のひとりだから作れる。周りに同じことをしている人がいなくて，周りに見てくれる人がいなければ，作品はできないってことです」
▶「うん，それはある。チームの中にいることで，触発されるものがあるもんね」
▶「僕の周りにはひとりもいませんけど，ひとりで作ってますよ」
▶「でも，Iさんの場合，何度もDōGAに遊びに来て，制作途中の作品なんかを見せているでしょ。それにCGAコンテストにも参加している。決して離れ小島という状態じゃない」
▶「離れ小島になっちゃっている人って多いと思う。そういう人がひとりでDōGAとコネクションを持つこ

読者通達事項

■第1回CGAシステム討論会のお知らせ

CGAシステムのこれからのバージョンアップのためにも，広く一般のユーザーの皆さんの意見をお伺いしたいと考えています。そこで，大阪近辺のユーザーの皆さん，当チームのスタッフといっしょにCGAシステムの問題点について夜を徹して語り合ってみませんか？（本当に夜中に開催するとは限りません）

第1回はCADを中心に，まったく新しいモデリング法からちょっとした改良点など，新しいCADの仕様書を公開しながら，和気あいあいと語り合いたいと思います。

月日は11月中旬を考えています。特にアイデアがなくても，CADを頻繁に使用している方なら結構ですので，参加ご希望の方は当プロジェクトルームの“第1回討論会事務局”までおハガキにてご連絡ください。のちほどこちらからご連絡いたします。

〒533　大阪市東淀川区淡路　5-17-2　篤コーポ102号
DōGAプロジェクトルーム

■CGAシステムマニュアル配布のお知らせ

CGAシステムがバージョンアップするにつれて，だんだんマニュアルの内容が食い違うようになってきました。そこで，現在のマニュアルの残りを処分したいと思っています。チームやクラブでCGAシステムを使っているが，マニュアルの数が足りなくて困っているというような方がいらっしゃいましたら，ハガキにてご連絡ください。実費でマニュアルのみの配布を行います。数に限りがありますので，申し込みは早いもの勝ちです。

■サイクロンCG大会のお知らせ

先月号のペンギン情報コーナーにて掲載されておりましたように，9月24日午後1時半より，渋谷道玄坂のフォーラム8にて，アンス・コンサルタンツ主催「第2回サイクロンCG大会」が開催されます。名誉なことに，DōGAを代表してかまたも審査員をさせていただいています。また，当日会場では「好きやねんCG　大阪合宿」で話題を呼んだZガ○ダムの変形シーンや，チームTOSAKAのグリフォ○なども公開する予定です。皆さんもぜひ，いらしてください。

■先月号の間違いのお知らせ

本誌9月号の本連載のモデラー高津のLOGINのコーナー内の“FFE”はすべて“FF”の間違いです。この場をお借りしてお詫びいたします。

とができると思いますか？　私は難しいと思いますよ」
——そのひとつの道として，CGAコンテストへの参加がある。ここに集まった人の多くもコンテストの参加がきっかけになってますから。しかしそれは作品ができなければ参加できないので，鶏か卵かということになってしまいますが。

▶「現時点では，悪い環境下に耐えて各自の奮起に頼らないといけない面があるのはしかたがないでしょう」

▶「各自ができる範囲で努力するというのがDōGAの基本方針だし」

▶「Oh!Xで各地のチーム紹介をやったでしょ。ああいうのも，どんどん続けてほしい」

▶「私のチームも掲載しましたが，ほとんど連絡がなかった」
——あの掲載で人数が集まったという例も聞いていますが，もちろん，成功といえるレベルじゃない。でも，私はあまり悲観していない。こういうことは，地道に続けていけばいいと思っています。あれは1回目ですから，とりあえず始めたということでお許しください。

▶「公開されている連絡先がDōGAだけなら，1局に集中するのは当たり前。あの掲載によって，少しずつでも，各サークルに連絡が入っているのなら，成功じゃないですか」

▶「私のチームはいまでも受け付けているから，どんどん連絡入れてほしい」

＊

……という具合に，話は各自の体力が続く限り果てしなく行われました。たいへん貴重なご意見も多く，なかなか有意義なイベントであったと思います。来年はさらに参加者を増やして行いたいのですが，金沢であるSF大会で行おうという案もあり，まだ未定です。皆さんも機会がありましたら，積極的なご参加をお願いします。

ビデオボードの善し悪し

シャープから7月に発売されたビデオボードを見て，まるでCGAシステムのためのボードのようだと思われた方も多いでしょう。私も以前から鳥居部長と会うたびに，S端子のビデオ出力を本体に付けてくださいとお願いしていました。しかし，新機種の本体に付けるとなると，以前のユーザーが対応できないので，ボードとして商品化するとのお話をいただきましたのはずいぶんと昔のことです。本連載において，一度もイメージユニットを使った録画方法を取り上げなかったのは，その辺の事情がありました。

しかしですね，確かに私はボードでもいいと言いましたが，まさか2スロット占有するなんてことは聞いておりません。EXPERTなど大部分の機種が2スロットしかない以上，増設メモリや数値演算プロセッサなどCGAに必須のアイテムが搭載できなくなってしまうではありませんか！

“好きやねんCG　大阪合宿”でも，このビデオボードは展示され，画質のよさは非常に好評でした。イメージユニットと比較して，その差はひと目見てわかるほどで，特に赤い色がにじまず，ゴーストが発生しなくなったのがうれしい点です。もちろん，ディスプレイと同等というわけにはいかず，ドットがしっかり四角形のまま録画できるというほどではありませんが，そのくらいの劣化は，CG独自のあのギザギザ感をカバーするのに適当で，アンチエリアスをかけた動画を録画した場合にちょうどよい具合になります。また，イメージユニットのように外部にビデオ信号を発生する装置（VTRなど）を接続する必要もなく，CGAのビデオ落としの作業がずっと手軽にできます。

しかし，“大阪合宿”に集まった方々も，2スロット使うというデメリットがあまりにも大きく，購入は見合わせたいとの感想がほとんどでした。せっかくのこの画質がもったいない！　なんとかよい方法がないものかとビデオボードをよく見てみると，実によい方法が浮かんできました。ちょっとした改造で，スロットに差す必要のない，名づけて“ビデオユニット”ができるのです。

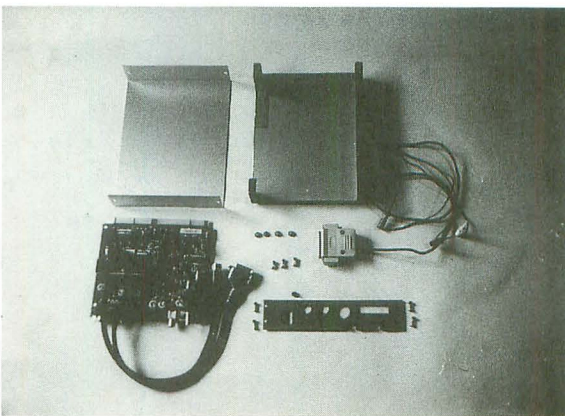
ということで，本連載始まって以来のハードウェア講座です。とはいっても，非常に簡単なものですので，ぜひ皆さん挑戦して，よい画質のCGAを楽しんでください。なお，この改造によってビデオボードが故障などした場合，DōGAも，Oh!Xも，シャープも責任をとってくれませんのでご注意ください。

ビデオユニットの製作

by　モデラー高津
製作費　約3,000円
難易度　☆

・制作方針

ビデオボードを見ればすぐにわかりますが，スロットについている端子はほとんどありません。調べてみると，つながっているのはグランド（0V）と+5Vと+12Vの3本だけでした。つまり，ビデオボードは電源をとるためだけにスロットを2つも占有しているのです。そこで電源を別なところからとり，ボードを手作りのケースに入れてしまえば，あっというまにスロットをひとつも使わないビデオユニットができてしまうのです。

別な電源として，+5Vと+12Vが出ているX68000本体の端子を捜してみると，ちょうどカラーイメージユニット用のイメージ入力コネクタがあります。ビデオボードを使用しているときにはカラーイメージユニットは使用しないでしょうから，電源はここからとれば問題ありません。

ケースの加工は，ビデオボードにスロットカバーがつきますので，それに合う長方形の穴と固定用の穴を4つあけ，ボード固定用の穴を3つあけるだけです。

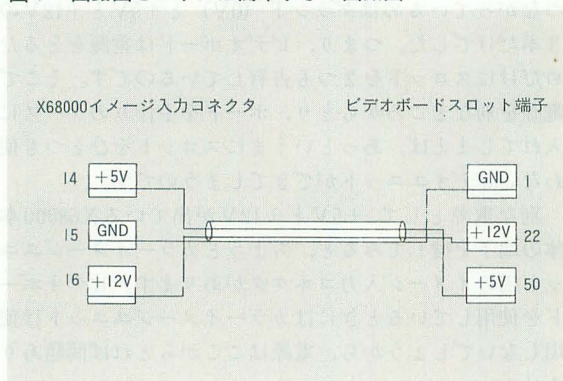
ハードの改造は初めて，という方もいらっしゃるでしょうから，ビデオボード自体の改造は行いません。みのむしクリップとICクリップを使ってビデオボードに接続します。要するに，改造といっても，電源ケーブルとビデオボードを入れる箱を作るだけなのです。

・部品

ビデオボードさえ入手できれば，あとは入手が難しい部品は特にありません。東京・秋葉原や大阪・日本橋に限らず，地方のパーツショップでも手に入ると思います(別コラム参照)。25ピンDsubコネクタ（X68000本体の背面のイメージ入力端子につながるもの）はメスです。ケースは，ビデオボードが入ればそれで問題はありません。2芯シールド線は見た目を気にしなければどんなコードでも結構です。

・部品一覧

品名	サイズ	数量	価格
アルミケース(例 タカチ YM-200)	150×200×40	1個	1050円
25ピンDsubコネクタ(メス)		1個	500円
〃 カバー		1個	395円
みのむしクリップ(小)		1個	20円
ICクリップ		2個	120円
2芯シールド線		1m	100円
スペーサー	4mm	3個	30円
ネジ	φ3×10mm	7個	50円

・工具

半田ゴテ，半田，ニッパ，ラジオペンチ，ドライバー，ハンドドリルはまず必要です。ケース加工にはハンドニブラーがあったほうが便利ですけど，ドリルで穴をあけてからやすりで削ってもよいでしょう。ドリルもないという方は，ネジではなくボンドでくっつけるなりしてください。

・製作

まず電源ケーブルを作ります。2芯シールド線は50cmもあれば十分でしょう。2芯シールド線の一方の端は，25ピンDsubコネクタの14番ピンに黒色（もしくは青や赤）の芯線を，15番ピンに網線（網状になっている導線を引っ張って，こよりのように線状にしたもの）を，16番ピンに白色の芯線を半田付けします。ピンの番号は，コネクタをのぞいてみるとちゃんと書いてありますが，左下の端が14番，そこから右に15，16番と続いています。

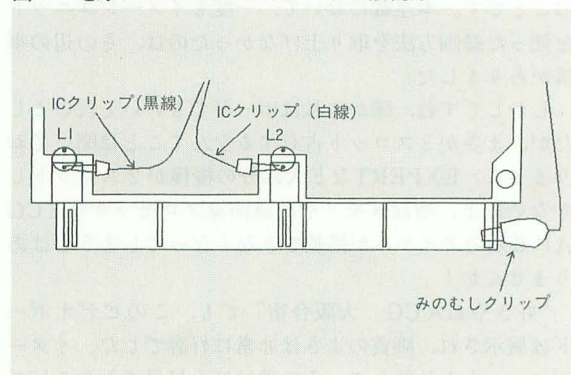
2芯シールド線のもう一方の端は，白・黒の芯線にはICクリップを，網線にはみのむしクリップをつけます(注意：カラーページの写真では，3つともICクリップになっています)。ICクリップへの半田付けは，まず，ボタン状になっている部分を強引に引っ張ってはずし，芯線をボタンの穴に通し，中の金属板に接続します。そして再びボタンを強引にはめ込んでください。みのむしクリップは，ゴムのカバーを引っ張ってはがし，そのカバーに網線を通してクリップに接続し，またカバーをかぶせます。半田付けはこれだけです。

テスターがある方は，X68000本体の電源を切ってから電源ケーブルだけを接続し，電源を入れてみて，電圧がみのむしクリップに対して（ベースにして），白色の芯線は12V，黒色の芯線は5Vであることを確認してください。電源が入らないときは，ショートしている可能性があります。確認してください。

大丈夫だったら電源ケーブルをビデオボードに接続します。接続は図を参照して行ってください。みのむしクリップはビデオボードのスロット部（差し込み口）をこちらに向けて，右端の1番ピンをはさみます。黒の芯線につなげたICクリップは左端の50番ピンにつながっているコイルL1（基板に小さくL1と書かれた100Kの黄色い部品）の2本の足のうち，スロット側の足に引っかけます。白の芯線のICクリップは，スロット部の真ん中辺のピン（22番ピン）につながっているコイルL2（基板にL2と書かれた100Kの黄色い部品）の2本足のスロ

図1　回路図というには簡単すぎる回路図

図2　電源ケーブルの実体配線図

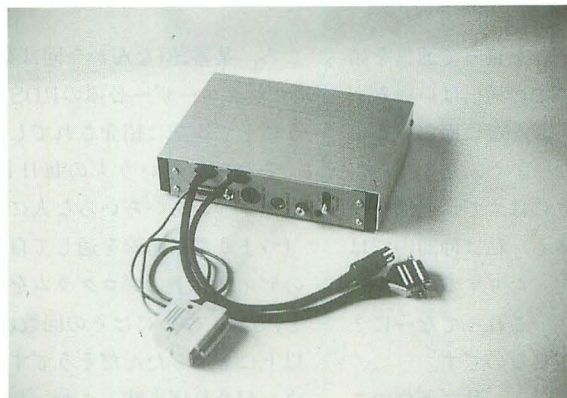
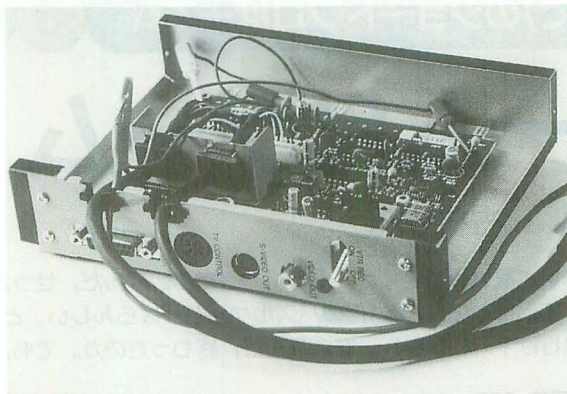
図3　電源ケーブルとビデオボードの接続部

ット側にL1と同様に引っかけます。

以上で配線は終了しました。簡単なので，10分もあればできると思います。ここで，一度X68000本体に接続して動作を確認しましょう。電源を切った状態で，取扱説明書にしたがって，各種ケーブルを接続します。本体の電源を入れてみて，ICが熱を持っていないことを確認しましょう。もしおかしかったら，ただちに電源を切って，配線が間違えてないか確認してください。問題がなかったら，なにかアニメーションを表示させ（15kHzモードにすることをお忘れなく。SRANIMの場合/Lオプションを付けてください。HANIMの場合，実行後Lキーを押してください）画面をTVに表示して，ビデオ信号が出ていることを確認します。

これで完成です。ケースを加工するのが面倒な方はこのままむき出しで使用したり，紙の箱に入れても結構ですが，ノイズを減らすためにも見た目のよいアルミケースに入れましょう。幸いにもビデオボードにはスロットカバーがついていますので，簡単にきれいなケースを作ることができます。アルミケースの後面にはスロットカバーがつくように155×40㎜の長方形の穴をあけます。ハンドニブラーがあれば簡単ですが，アルミなので，ニッパでできるでしょう。その穴の両隣にスロットカバーのねじ穴に合うように直径3.2㎜の穴を4つあけます。ケースの底にはビデオボードの3つの穴にあうように穴を開けます。これでケースの加工は終わりです。

ケースの底に4㎜のスペーサーをはさんでビデオボードを固定し，後面にスロットカバーをつければ完成です。なにしろアルミのケースですので，ボードを直接置くとショートしてしまいます。必ず，ボードの裏がアルミに接しないように注意してください。不安でしたら，厚紙か何かをビデオボードの大きさに切って，ケースの内側に貼ってください。また，アルミケースはグランドにちょうどよいので，みのむしクリップで摘んだところとアルミケースを，適当につなげてやることをお勧めします。

電源ケーブルもスロットカバーから出します。電源ケーブルをX68000本体のイメージ入力コネクタに接続する以外は，使い方はビデオボードとまったく同じです。

（注）コネクタの抜き差しは電源を切っているときにしましょう。電源を入れたままコネクタを差すと，いきなり電源が落ちる場合があります。その場合は慌てずに一度電源を切れば大丈夫です。

終わりに

連載をせっかく隔月化にしてもらったのに，間に緊急レポートを入れていたんでは意味がないような気がします。

ところで，この原稿のために友人（高校の同級生）の結婚式に行けなくなりました。池田君，あゆみさんご結婚おめでとうございます。学生結婚ということで，いろいろ苦労もあるでしょうが，どうせ結婚なんて，愛と思い込みと勢いがないとできないものでしょうから，私は賛成です。いつまでも愛を貫いてください。

さて，来月号はMAX田口君による「宇宙要塞CADの逆襲」です。私は季節外れの旅行でしばらく不在になりますので，すべて田口君に任せています。多少不安ですが，お楽しみに。

ビデオユニット改造パック通信販売のお知らせ

ハードの改造など初めてという方は，部品を揃えるのにもいろいろ不安があるでしょう。25ピンDsubコネクタといってもどんなものかわからないし，ちょうどよい大きさのアルミパックが見つからないかもしれません。

そこで，前記部品一覧の値段を調べるついでに，通信販売をしてくれるお店を見つけておきましたので，自分で部品を揃える自信がない方はご利用ください。右記の住所に現金書留で申し込んでください。なお，この通信販売はDōGAは関与しませんので，お問い合わせは直接お店のほうへお願いします。

申し込み先　〒556　大阪市浪速区日本橋5-7-19　共立電子産業（株）
TEL　06(644)4446

申し込み方　「Oh!X ビデオボード改造パック」と記入のうえ，前記の部品一覧を添える。

値段　近畿地方の方　3,000円（消費税，郵送料込み）
その他の地域の方　3,500円（同上：地域による郵送料の差額分は切手として返送されます）

★(で)のショートプロぱーてぃ その14

ゲーム&ゲーム

Komura Satoshi 古村 聡

さあ，いよいよ夏休みだ。えっ，もう終わったの。なんだ，せっかくゲームを2本も用意してあげたのに，X1用のパズルゲーム「ばらんしい」とX68000用ゲームの「それ行け！ ロケット」を。なんだ，終わったのか。でも，やってね。

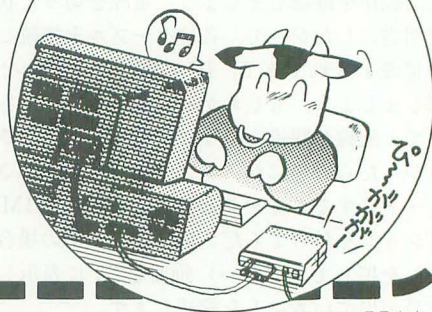



illustration : T.Takahashi

はろー。ついにモデムを買って通信を始めてしまった（で）です。通信はいいぞぉ，楽しいぞぉ。いやー，こんなに面白いものだとは思わなかったなー。ぐふぐふ。

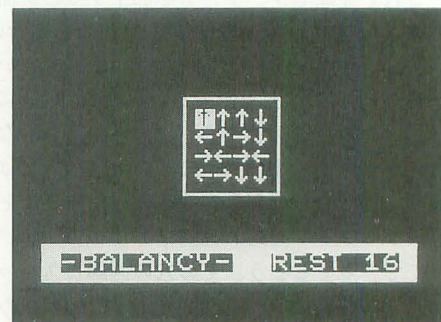
実をいうと私の周りにはすでに通信をやっている人はかなり多く，私は仲間内ではかなり出遅れたほうだったりするので，ほかの人にあれなーに？ これってなーに？と聞きまくっている毎日なのです。

で，そんなある日のこと，私に通信のことを教えてくれるM氏がこんなことを言ってました。

「PDSのレスって全然ないんだよね」

このM氏，その名のとおりバリバリの某舶来機Mのユーザーで，プログラムなんかもガシガシ強力なものを組む人だったりするのですが，彼の組んだあるプログラムを某大手ネットにPDSとしてアップしたところ，某雑誌(なんか今回は某が多いな)に「マックユーザー必須のPDSソフト」ということで雑誌に紹介されてしまったのです。

それからというもの毎日毎日そのPDSはネットからいろいろな人にダウンロード（つまり電話回線を通して自分のパソコンのディスク上にプログラムをコピーすること）され，ついにはその回数は1週間で100以上にもなったんだそうです。それだけダウンがあれば当然，よかったよー，とかちょっとここは直したほうがいいんじゃないとか，レスポンスがいろいろ来るだろうと思いきや，実はたったの1通しか来なかったんだそうで，M氏はとてもがっかりしていました。

雑誌でPDSのことがよく「ただで使える便利なソフト」という感じで紹介されるためなのか，ただでもらえるソフトはもらっちゃえ，もらったらそのまんま，もらっちゃえばこっちのものだというような人が多いみたいなんですね。初心者がこんなことをいう筋合いではないかもしれませんがはっきりいって，私は悲しいです。

ばらんしい

プログラムっていうのは人が苦労して作るものだし，ましてやPDSに至ってはほとんどボランティア。作る人も使った感想

リスト1 ばらんしい

```
1 '-BALANCY-   Version 1.18   '90/6/24
2 '     (C)1990  Sakamoto Yasushi
10 WIDTH 40:CLS4:INIT:CSIZE3:"FADE"
20 LINE(0,0)-(319,199),PSET,1,BF
30 DEFINT A-Z:DIM B(9,9),BS(9,9),C$(4)
40 F=4:FORI=1TO4:C$(I)=CHR$(27+I):NEXT:C$(0)=" "
50 '
60 FORI=0TOF+1:FORJ=0TOF+1:BS(I,J)=0:NEXT:NEXT
70 FORI=0TO3:FORJ=0TOF^2/4-1:K=I*F^2/4+J
80 BS(K¥F+1,K MOD F+1)=I+1:NEXT:NEXT
90 FORI=1TOF:FORJ=1TOF
100 SWAP BS(I,J),BS(INT(RND*F+1),INT(RND*F+1)):NEXT:NEXT
110 FORI=0TOF+1:FORJ=0TOF+1:B(I,J)=BS(I,J):NEXT:NEXT
120 CLS:LOCATE18-F,8-F:PRINT#0"┌";STRING$(F,"─");"┐"
130 LOCATE18-F,F+10:PRINT#0"└";STRING$(F,"─");"┘"
140 FORI=1TOF:LOCATE18-F,8-F+2*I:PRINT#0"│";SPC(F);"│":NEXT
150 LINE(0,20)-(39,23),"■",BF:LOCATE0,20:PRINT"■"
160 R=F^2:LOCATE2,21:PRINT#0USING"-BALANCY-■■REST ##",R;
170 FORQ=1TOF:FORP=1TOF:"PRT":NEXT:NEXT:"F-IN":X=1:Y=1
180 LOCATE34,21:PRINT#0USING"##",R;:IF R=0 GOSUB"???":GOTO60
190 CREV1:P=X:Q=Y:"PRT":CREV:PAUSE1
200 S=ASC(INKEY$(0)):IF S=27 THEN "MENU"
210 S=S-48:IF S>9 THEN S=S-96:"CHANGE":GOTO180
220 M=(S=6)*(X<F)-(S=4)*(X>1):N=(S=2)*(Y<F)-(S=8)*(Y>1)
230 IF M+N=0 THEN 200 ELSE P=X:Q=Y:X=X+M:Y=Y+N:"PRT":GOTO190
240 '
250 LABEL"PRT":LOCATE18-F+P*2,8-F+Q*2:PRINT#0C$(B(P,Q)):RETURN
260 LABEL"CHANGE"
270 M=(S=4)*(X<F)-(S=5)*(X>1):N=(S=2)*(Y<F)-(S=3)*(Y>1)
280 IF M+N=0 OR B(X,Y)=0 OR B(X+M,Y+N)=0 THEN 310
290 SWAP B(X,Y),B(X+M,Y+N):P=X:Q=Y:"PRT":P=X+M:Q=Y+N:"PRT"
300 U=X:V=Y:"CHK":U=X+M:V=Y+N:"CHK"
310 RETURN
320 LABEL"CHK":C=B(U,V):PP=(C=2)-(C=1):QQ=(C=3)-(C=4)
330 IF C=1 AND B(U+1,V)=2 GOSUB"ERS"
340 IF C=2 AND B(U-1,V)=1 GOSUB"ERS"
350 IF C=3 AND B(U,V-1)=4 GOSUB"ERS"
360 IF C=4 AND B(U,V+1)=3 GOSUB"ERS"
370 RETURN
380 LABEL"ERS":FORI=7TO0STEP-1:COLORI
390 P=U:Q=V:"PRT":P=U+PP:Q=V+QQ:"PRT":NEXT:COLOR7
400 B(U,V)=0:B(U+PP,V+QQ)=0:R=R-2:RETURN
410 LABEL"MENU":"FADE":LINE(0,0)-(39,19)," ",BF
420 LOCATE14,4:PRINT#0"-MENU-":LOCATE6,8:PRINT#0"●  TRY AGAIN"
430 LOCATE12,12:PRINT#0"MODE CHANGE":Q=0:"F-IN"
440 WHILE STRIG(0)=0:S=STICK(0):Q=Q-(S=8)*(Q>0)+(S=2)*(Q<1)
450 PRINT#0" ":LOCATE6,Q*4+8:PRINT#0"●":LOCATE6,Q*4+8:PAUSE1
460 WEND:IF Q=0 GOSUB"FADE":GOTO110
470 WHILE STRIG(0):WEND:REPEAT
480 S=STICK(0):F=F-(S=4)*(F>4)*2+(S=6)*(F<8)*2
490 LOCATE22,12:PRINT#0USING"#×#   ",F,F:PAUSE1:UNTIL STRIG(0)
500 "FADE":GOTO60
510 LABEL"FADE":FORI=0TO7:PALET 1,I:PALET 0,I:NEXT:RETURN
520 LABEL"F-IN":FORI=7TO0STEP-1:PALET 0,I:PALET 1,I:NEXT:RETURN
530 LABEL"???":PAUSE10:"FADE":P$="⌒＼|╭ ╰●=━▄█▒└┬┼▲"
540 CREV1:CLS:RESTORE 610:FORI=0TO10:READ D$:FORJ=1TO LEN(D$)
550 C=ASC(MID$(D$,J,1)):IF C<65 THEN PRINT#0SPC(C-48);:GOTO580
560 IF C>84 THEN CREV
570 PRINT#0MID$(P$,C-64,1);:CREV1
580 NEXT:PRINT#0:NEXT:PRINT#0"  CONGRATULATIONS!";:CREV
590 FORI=6TO0STEP-1:PALET 1,I:NEXT
600 WHILE STRIG(0)=0:WEND:FORI=1TO7:PALET 1,I:NEXT:CLS:RETURN
610 '
620 DATA 7AC2CB,6CIC2CIC,4KKRJRKKRJRKK,4VLT2MU2TLU,6CDJEDJEC
630 DATA 6CCHCCHCC,5DPFJGFJGOE,5C2NNNN2C,5C8C,5FJJSJJSJJG,8FJJG
```

が聞きたくて作った人も少なくないはずですし，人にプログラムの評価をしてもらってうれしくない人はいないですよね。

「タダで配ってんだからいいじゃん」などといわずに，ダウンロードしたらちゃんとレスを返してあげましょう，という(で)の意見でありました。じゃんじゃん。

白と黒は4倍角のしるし！

さてと，余談はこれぐらいにして今月の1本目いきましょう。今月の1本目の作品はX1用のまたまたパズルゲームで「ばらんしぃ」です。

ばらんしぃ for X1 シリーズ
(CZ-8FB01)
秋田県　坂本康

またしても，恐怖の4倍角男・坂本さん（ごめんなさい）によるX1用のパズルゲームです。

画面上に矢印が並んでいますね。この矢印を→←のように向かい合わせると互いに打ち消し合って消えていきますので，そうやって画面上の矢印をすべて消していってください。あ，

←→

や，

→↓
↑←

では消えませんので注意注意（それじゃあパズルにならんもんね)。そんで，カーソルの移動方法ですが結構面倒臭いのでよく聞くように。うんとですねえ，まず，2，4，6，8のキーでカーソルが下左右上に動き，GRAPHキーと2，4，6，8キーを押すと，その方向の矢印とカーソルのある矢印とが入れ替わります。あとESCキーでメニュー画面。メニュー画面では2，8でTRY AGAINとMODE CHANGEの選択，MODE CHANGEでは4，6で選択，スペースキーで決定です。

今度のパズルも死ぬほど手強いですね。先のことを考えてながらやっていると，頭がだんだんこんがらがってしまいます。しかし、面クリアしたときがなかなか「ぐふぐふ」であったりするので皆さんがんばってクリアしましょう！　ぐふぐふ（あんまりいっちゃうとバレそうだなあ)。しかし，キャラクタ画面でよくここまで描きましたね。2重丸あげよう。ぽん。

さて，作者の坂本さんは受験生ということでこのゲームを最後にX1を封印してしまうんだそうです。そうかそうか，もうそんな季節がやってきたか。懐かしいなぁ。しかし，受験生のみんながいなくなってしまうと私もさびしいな，うーん。ねえねえ，坂本さんもさ，いいじゃん受験なんて堅いことといわずにさ，ね，ね（なーにが，「ね，ね」なんだか……)。

まあ，冗談はともかくとして受験生の皆さん，がんばってくださいね。ご武運をお祈りしております。

飛んでけ，飛んでけ，青い空へ

さて，お次は今月の2本目（最近は月2本が完全に定着したな）はX68000用のゲームで「それ行け！　ロケット」です。

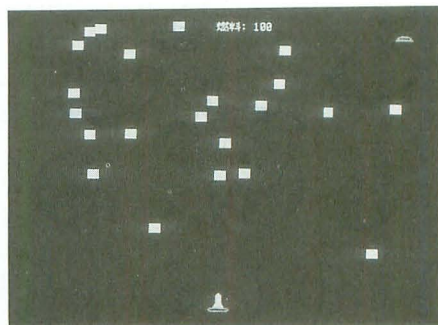
それ行け！　ロケット for X68000
(X-BASIC 要 sp_chk())
神奈川県　野田敏之

そんで，操作法&ルールになるわけですが，要するに障害物を避けてロケットをUFOにぶっつけちゃえば万事OKなわけです。ジョイスティックで台車を左右に動かしてAボタンでロケットを上向きに加速していき，離すと重力に引っ張られて落ちていっちゃう。ま，早い話がロケットには上に慣性が，下には重力がかかってしまうんだなこれが，もう。

おっとっと。一度台車を離れるとロケットは左右に移動できなくなるので注意するように。もうひとつ，燃料は少ないから無駄使いしないように。ほんじゃあ，がんばってUFOを撃退してちょ，てなもんです。

ほっほお。がんばりましたね。当たり判定にsp_chk( )が，そしてスプライトのデータには圧縮ルーチンが使われているんですね。大変よくできました。まる。

それでですね，投稿原稿によれば，「ロケットとUFOのヒットチェックにsp_chk( )を使っていますが，実際のヒットチェックはUFOのキャラではなくその後ろにあるダミーのスプライトに対して行っています。これによって視覚的に納得のいくヒットチェックになったと思います」だそうです。

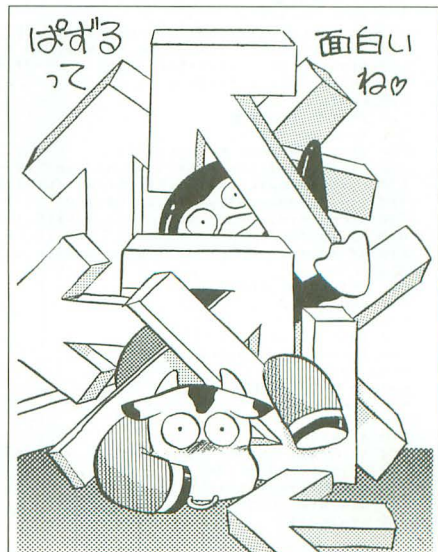
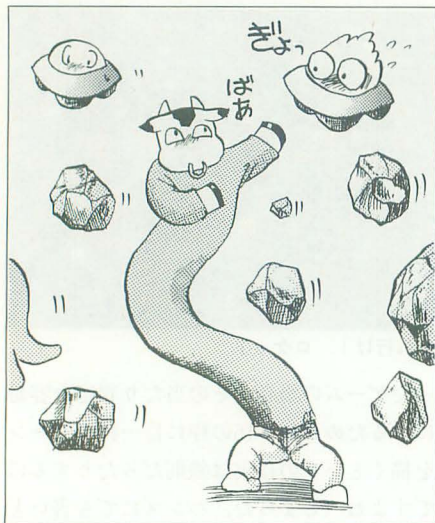
うーん，そうですねえ。sp_chk( )の仕様を見ればわかるように，スプライトを使

それ行け！　ロケット

ったゲームの場合，その当たり判定を容易にするために16×16の枠に目一杯パターンを描くというのは実は鉄則だったりするのですよね（しまった，ハンズにでも書いときゃよかった……)。その点，ダミーのスプライトをヒットチェックに使うというのはなかなかのアイデア賞ものです。

sp_chk( )の場合，対象になるスプライトがひとつだろうといくつになろうとそんなに変わらないので（BASICの文法チェックに一番時間がかかっていてそれに比べると実はスプライトのチェックなんて比較にならないくらい速い時間なのだ）これでもOKでしょう。

ちょっとお手軽な圧縮の話

ところで，ここで使われているデータ圧縮は，俗にいうランレングス法ってやつなのですね。ご存じない方にこの解説をしておくと，早い話，同じデータがいくつ続いているかを調べてデータを縮めていく方法なんですね。たとえば，

05，05，05，00，01，01，01，01

というデータがあったとしますね。こいつを，1個目がデータの中身，2個目がその繰り返し回数……として，

05, 03, 00, 01, 01, 04

と書いてやるんです。ほら，ここでデータが2個減ったでしょ。これをデータの個数分繰り返してやれば結構データが減るんじゃないかな，って寸法です。

懐かしいなぁ。実は私もマシン語で初めてプログラムを組んだときは，こいつで絵のデータを圧縮したんですよね。ただ，このプログラムの場合，あんまりデータが縮んでませんね。データを縮めるのに命を賭けちゃうんであればここは一発，スプライトを単色にして一気にデータを1/8にしてしまうとか，

（たとえば，

0, 3, 0, 3, 3, 3, 3, 3

これを3を1に見たてた2進数にする。

0, 1, 0, 1, 1, 1, 1, 1

それを16進数化して，

5f

と書くようにするとか……）

さらに上のパターンとのXORをとって0を縮めるとか……。面倒臭いか。

なんだかんだいっても，このランレングスっていうのは圧縮，展開のプログラムが簡単に組めるし（つまりデータの展開にそ

リスト2 それ行け！ ロケット

```
 100 /*  それ行け！ ロケット  FOR X68000 X-BASIC（要 sp_chk(
))  1990年7月6日（金）
 110 /*  PROGRAMMED BY TOSHIYUKI "みずのなかのわぁくぅせいまぁ
でぇー" NODA
 120 /*初期設定 1
 130 screen 1,1,1,1:console ,,0
 140 int mo,x,y,a,b,st,ux,hx,sx,sy,i,j,k,l,lo=498:float ry:char
 fn,en,ro,po,te,c(255)
 150 str D(5)[250],E
 160 sp_init():sp_on():sp_disp(1)
 170 sprite_pattern()
 180 /*タイトル
 190 while 1
 200 locate 23,11:print "それ行け！  ロケット
 210 locate 25,20:print "1990 material":sp_off():ro=0
 220 while strig(1)=0
 230 endwhile
 240 repeat
 250 cls:fn=0:ry=0:a=0:ro=ro+1:en=101:mo=1:x=200
 260 locate 23,15:print "ROUND ";ro
 270 vpage(0):sp_off():scrinit(ro*5+10):cls:en_print():vpage(3)
:sp_on()
 280 /*台車＆UFO移動
 290 while a=0
 300 a=daisya():ufo():scroll()
 310 endwhile
 320 /*ロケット変数設定
 330 x=x+8:y=476:a=0
 340 /*ロケット＆UFO移動
 350 while a=0
 360 ufo():a=rocket():b=hitchk()
 370 if b then a=28
 380 scroll()
 390 endwhile
 400 if a=28 then {sp_move(0,x,y,5):sp_move(1,x,y+16,5)
 410 locate 21,15:print "GAME  OVER"} else {
 420 sp_move(5,ux,16,5):sp_move(6,ux+16,16,5)}
 430 for i=0 to 2:apage(i):wipe():next:cls
 440 color 3:until a=28
 450 endwhile
 460 /*台車移動
 470 func daisya()
 480 sp_move(3,x,493,3):sp_set(4,x+32,509,&H4103)
 490 sp_move(0,x+8,476,0):sp_move(1,x+8,492,1)
 500 st=stick(1):x=x+2*(st=4 and x>0)-2*(st=6 and x<476)
 510 return(strig(1)=1)
 520 endfunc
 530 /*ロケット移動
 540 func rocket()
 550 if y>476 then return (28)
 560 st=strig(1):st=-st*(st<2 and en<>0)
 570 ry=ry+0.5#*(st=1)-0.5#*(st=0):fn=-(st=1)
 580 y=y+ry:ry=-ry*(y>-1):y=-y*(y>0)
 590 sp_move(0,x,y,0):sp_move(1,x,y+16,1+fn)
 600 en=en-fn
 610 if en then en_print() else locate 25,13:color 6:print "燃
料切れ！"
 620 return(sp_chk(0,7,7))
 630 endfunc
 640 /*燃料表示
 650 func en_print()
 660 locate 26,0:print "燃料:";en-1
 670 endfunc
 680 /*当たり判定
 690 func hitchk()
 700 hit=0
 710 apage(0)
 720 if x+511-sx>511 then hx=x-sx else hx=x+511-sx
 730 hit=hit+point(8+hx,y)
 740 hit=hit+point(4+hx,y+16)
 750 hit=hit+point(12+hx,y+16)
 760 apage(1)
 770 if x+sy>511 then hx=x+sy-511 else hx=x+sy
 780 hit=hit+point(hx+8,y)
 790 hit=hit+point(hx+4,y+16)
 800 hit=hit+point(hx+12,y+16)
 810 return(hit)
 820 /*UFO移動
 830 func ufo()
 840 sp_move(5,ux,16,4):sp_set(6,ux+32,32,&H4104)
 850 sp_move(7,ux+8,8,6)
 860 ux=ux-3*mo
 870 if ux<4 then mo=-1 else if ux>471 then mo=1
 880 endfunc
 890 /*障害物セット
 900 func scrinit(k;char)
 910 for i=0 to 1
 920 for j=0 to k
 930 apage(i)
 940 symbol(rnd()*700,(rnd()*400),"■",1,1,1,14,0)
 950 next:next
 960 endfunc
 970 /*障害物スクロール
 980 func scroll()
 990 home(0,511-sx,0):sx=sx+1
1000 home(1,sy,0):sy=sy+1
1010 if sx=512 then sx=0
1020 if sy=512 then sy=0
1030 endfunc
1040 /*キャラクタ・セット
1050 func sprite_pattern()
1060 D(0)="X0733X0E33X0DX34X0BX36X09X38X08X38X08X38X08X38X08X38
X08X38X08X38X08X38X08X38X08X38X
1070 D(0)=D(0)+"08X38X08X38X04
1080 D(1)="X04X38X08X38X08X38X08X38X07X3AX06X3AX05X3CX04X3C000X
3E00X3EX0FX0FX0FX0FX0FX0FX04000
1090 D(2)="X04X38X08X38X08X38X08X38X07X3AX06X3AX05X3CX04X3C000X
3E00X3EX0554544555X08C4D444D5X0
1100 D(2)=D(2)+"8D55DD554X08CC5544D4X09DCCDD4X0B44C4X0400
1110 D(3)="X0FX0FX0FX0FX0FX0FX0FX0FX09X8F8X04X9C0009888X0B9X
85X0A9X850088088X04988800088088
1120 D(3)=D(3)+"X05X94X94X94
1130 D(4)="X0FX0FX0FX0FX0EX56X08X58X07X59X0655544454454X06555445
4454X05X544454454X05X5BX04X5C00
1140 D(4)=D(4)+"0X5DX05X4400X44X0744X0444X0F000
1150 D(5)="50050C00040504C050C0504040050054 4X0540445C40C04505C0
5D040C455044D5054004500050400DD
1160 D(5)=D(5)+"00D0505D00044
1170 D(5)=D(5)+"0054550D4D00C050DDD5DDD0D0D0054D0D040005D05DC54
DD040D4X04D00C0D040DD0055D000C4
1180 D(5)=D(5)+"DD40D5XD55000450C05DD5D5000504 5000C045D0055500X
44044044C000CC004000C000554CC
1190 for l=0 to 5
1200 kaitou()
1210 next:endfunc
1220 /*かいとう
1230 func kaitou()
1240  po=1:i=0
1250 while i<>256:locate 23,11:print lo:lo=lo-1
1260 if mid(po)<>"X" then c(i)=val("&h"+mid(po)):i=i+1:po=po+1
else {
1270       te=val("&h"+mid(po+1)):j=val("&h"+mid(po+2))
1280        for k=1 to j
1290         c(i)=te:i=i+1
1300        next
1310       po=po+3}
1320 endwhile
1330 sp_def(l,c,1):sp_move(0,0,0,l)
1340 endfunc
1350 /*みっどだらー
1360 func str mid(po;char)
1370 return(mid$(D(l),po,1))
1380 endfunc
```

んなに時間がかからない),改良もしやすいんで，結構お手軽ないい圧縮の方法なんだよな，と私は密かに思ってしまいます（上級者の人には笑われるだろうけど)。が，もっと縮めたーいとおっしゃる方には，さらなる圧縮の方法もあるわけですが，それは皆さん各自で調べていただきましょう。たぶん参考文献も結構あるだろうし,Oh！Xのバックナンバーにもあるんじゃないかなと思います。がんばってね。

うーむ，ほとんど圧縮の話ばかりになってしまった。

さて，とりあえず今月はこんなもんですかね。投稿者の皆さんががんばっているので私はうれしい。これからもどんどん送ってくださいね。では，そんなこんなでまた来月。

さあ，また泥沼チャットでもやるか！

(で)のぱーてぃハンズ――――(番外編)

おなかがへったよぉ

ぴーひゃら，ぴーひゃら，……。はい，皆さんこんばんは。ぱーてぃハンズの時間がやってまいりました。

さて，先月をもちまして無事，第1部が完了したぱーてぃハンズ，今月は質問のハガキでも来たら，なんか質問箱でもやろうかな……と思って1回空けておいたんですけど（決して第2部のネタがなくて安易にそうしたわけではない,ないってば)。なんだなんだ，待てど暮らせど来やしないハガキなんか。

まあ，このぱーてぃハンズ，元々がたいして内容がなかった（うっ）うえに，6カ月もかけてちんたらちんたら書いたから，ひょっとしてもう読まれてないのかなあ……なんて心配もしてしまうのですね。これが。まあ，いいや。質問がないんだったら適当に無駄話でもしましょう（いいかげんだなあ)。

やっぱりパソコンの話なのね

最近，PDSとか同人ソフトとかが流行ですよね。最近，通信始めたおかげで思うのですが，ま，アクセスしてるネットにかなり偏りがあるんで私の感想もちょっと偏っているかもしれないけど……，X68000のPDSってすごく豊富なんですね。びっくりしてしまいました。私自身はあまり難しいことはわからんので（GNU CとかJgawkぐらいしかわからんの)，ゲーム関係の話になってしまいますが，これがすごいっ！

昨日，NIFTYからOh! HAJIKIというゲームをダウンロードしたんですが，こいつがすごいったらすごいっ！　BGMバリバリのサンプリング音ガシガシでもう市販ゲームとしても売れるんじゃない？　というぐらいの出来なんです。

それに先週ダウンしたので，すごかったのが（ゲームじゃないんだけどね）POLI3！　3Dグラフィックが飛んで飛んで回って回るプログラムです。

いやあ，X68000ユーザーってすごい。ユーザーの盛り上がりがすごいし（PDSだけじゃなくて普通の書き込みもすごい量！　3日読まないとあとで読むのが大変)，プログラムもガシガシ出てくる。これで出荷台数がPC-9801の1/10っていうのはいったいどういうこと!?　と思ってしまいます(PC-9801はもっとすごいのかな)。

もう私はこうなったら宣言してしまいます。

X68000は文化なのじゃあ!!!

プログラムの組めるそこの君！　君こそ文化の担い手なのだ！　さあ，がんばってX68000を盛り上げてくれたまえ。

んでもってハンズの宣伝ね

さて，問題はプログラムが組めなくてギャラリーしている君だったりするのだな。これこれ逃げるんじゃないって。

うん，確かに最近，プログラムしない人って増えてるみたいなんですよね。アンケートハガキにも一番良かった記事＝SOFTOUCH（ソフトのレビューのページ）っていうのが結構ありますからね。それはそれでいいんですけど。

でも，私はそういう人にいってあげたいのだ。

プログラムってのも面白いんですよぉ！

たぶんですね，そういう人って，“ひとつぐらい自分で作ってみたいんだけど，なにをどうしていいのかわかんない”んだと思うんだな。きっと“自分にはゲームなんて作れっこない”と思ってるんだな。

そんなことないんです。よくいわれるセリフだけど“誰だって最初は初心者だった”んだから。かくいう私だっていまでこそパソコン雑誌でいろいろ偉そうに書いてますけど（まあ，私程度の力の者がいっても説得力ないのはわかってますけど)，私もね，最初はゲームオンリーの人だったんです，実は。

あれはいまを遡ること×年前。当時，純真な紅顔の美少年だった私（自分で書いても恥ずかしいな，これ）は自宅でゲームしたいがために買ってしまったんです，パソコン。もちろん親には「コンピュータが勉強したいから！」とかなんとかいってだまくらかしてたけど，とーぜんゲームがしたいからに決まってんじゃん，という純真な少年でした。

いや，燃えましたねえ。ゲーム。特に，なんといっても燃えたのはあのマッピー。ああ，あの愛らしいマッピーやらニャームコやらのキャラクタたち，そしてトランポリンを使う斬新なアイデア！　指と指の間が皮がムケて血が出るまでやったおかげでマッピーだったら誰の挑戦でも受ける！　というところまでやり込んだものです。

そこで，古村少年，ムクムクと頭の中で「こういうゲームが作ってみたい！」という誰もが一度は抱く野望をやっぱり抱いてしまったのですねえ，これが。

が，私ははっきりいってマッピーばっかりやっていてBASICすらよく知らなかったんですよ，そのとき。プログラムを組んでゲームを作れることは知ってたんですが，遊んでばっかりでBASICの文法なんてほとんど知らなかったんですよね。知っているのはRUNとかLISTとかのコマンドぐらいでステートメントは全然わからん，という困った奴であったのです，実に。

で，とりあえず，参考書を買ってきたんですが，よくわかりませんでした。そこで，そのへんのサンプルプログラムをわけもわからず打ち込んでみたんですね。いや，動きましたよ。画面の真ん中でXが右に左に。感動しましたよ。そんで，そこから少しずついろんなものを本で読んだり，人に教えてもらったりでいろいろ憶えていって，いま私はOh！X編集室でご飯を食べさせてもらってたりする，ということになるわけです，はい。

はっきりいって，プログラムなんて要領です。最初のとっかかりさえなんとかなれば，あとは勢いで結構いっちゃうものなんですよ。だから，食わず嫌いしないでとりあえずはBASICでもかじってほしいなー，と思う私なのでした。

幸い清水さんが先月から新連載，泉さんのX-BASICが来月から新展開，そして不肖わたくし（で）のぱーてぃハンズも来月から第2部に入り，入門用記事揃い踏みですんで，

みんなで文化しようぜ！

と叫んでしまう私なのでありました。

いやあ，久しぶりに好きに文章を書いて，ひとりで満足している私なのでした。うーん，気持ちいいなあ。

では，また来月Oh！Xでお会いしましょうね。よいしょぉ！（とX68000を抱えて去る）

シナリオ作成の支援

短期集中連載ということで，ポケコンを使ってテーブルトークRPGを楽しむためのプログラムを紹介してきました。最終回は，AD&DおよびD&D用の細々としたツールを発表し，あわせてシナリオの作り方にも触れておきましょう。

Matsui Shin
松井 信

先月のCSTは試していただけましたでしょうか。今月は先月載せきれなかったツール類を紹介しておきます。

シナリオ支援ツール

●リスト1「thac0.2##」

モンスターやキャラクタのTHAC0を計算します。実は前回のCSTのサブルーチンを抜き出したものですが，RAMファイルに入れておきキー一発で出るようにすると，わざわざ表を引かないですむので便利です。なお，AD&D (2nd edition)用なので，D&Dの人は前回の変更部分を参考に自分で書き直してください。使い方は，モンスターのHDをそのまま入力する（たとえば，トロールなら「6+6」）か，たとえばレベル7ファイターなら「F7」と入力すれば，THAC0が表示されます。

●リスト2「dice.3##」

これも前回のプログラムのサブルーチンで，一種の電子サイコロです。たくさんのサイコロや24面などの変則サイコロを振るときに重宝します。ndm＋pをn＊m＋pとして入力すれば（たとえば，16d6＋16なら，16＊6＋16）乱数で結果が出ます。これもRAMファイルに入れておき，キー一発で出るようにすると便利です。

これらのプログラムは，ゲーム中はもちろんのこと，シナリオ作成のときに便利です。私は多くの場合シナリオを電車の中とか授業中(！)とかに作っているのですが，いちいちマニュアルを持ち歩くのは面倒です。それに考えてみるとマニュアルはほぼ暗記しており，必要なのは表の部分だけなのです。というわけでこのプログラムを作りました。こういったときにポケコンの機動性が発揮されます。

シナリオ作成の目安

初心者のマスターはまずモジュールをやれといいますが，モジュールは高いし，そのうちやっぱり自分でシナリオを作りたくなります。というわけで，シナリオメイクのHow toについて若干ふれておきます。

●ダンジョンダンジョン

ダンジョンといえばうっかり土牢などと訳されたものを見かけますが，RPG用語ではモンスターや宝がちりばめられた迷路を指します。このダンジョンがシナリオにあればとりあえず間が持つので，RPGの必需品といえましょう。

というわけでダンジョンですが，注意すべきは「広すぎず狭すぎず」というところです。RPGの楽しさの中で「異世界を探検する」というのはかなり重要で，小さすぎるダンジョンは物足りなさを感じてしまいます。かといって，広すぎるダンジョンも困りもので途中でだれてしまいます。RPGのシナリオというのは要するにエンターテイメントなのですから，いかに盛り上げるかというのが大事になるのです。

エンターテイメントの演出として重要なことは，

1) いきなり盛り上がりを持ってきてはいけない

盛り上がりのピークを高めるためには，その前にウォーミングアップが必要。ピークの前にひとつ小さな山を作っておくとピークをさらに押し上げることができる。

2) ウォーミングアップが長すぎない

前置きが長すぎると刺激に対する感受性が鈍くなり，ピークを逆に下げてしまう。

3) 盛り上がりのあとも大切に

終わったあとに爽快感を持たせるためと，

リスト2 DICE.3##

```
100 RANDOMIZE
110 CLS
120 PRINT "nDm+p ... n*m+p"
130 INPUT "?";DD$
140 D$=DD$
150 N=VAL D$
160 *LOOP
170 D$=RIGHT$ (D$,LEN D$-1)
180 IF LEFT$ (D$,1)<>"*" THEN *LOOP
190 D$=RIGHT$ (D$,LEN D$-1)
200 D=VAL D$
210 P=EVAL D$-VAL D$
220 FOR J=1 TO N
230 P=P+RND D
240 NEXT J
250 PRINT ".......";P
260 GOTO 130
```

リスト1 THAC0.2##

```
100 CLS
110 PRINT "** THAC0 CALC **"
120 INPUT "HD/CL+LV=";HD$
130 M=1
140 IF LEFT$ (HD$,1)="F" THEN M=2
150 IF LEFT$ (HD$,1)="C" THEN M=3
160 IF LEFT$ (HD$,1)="T" THEN M=4
170 IF LEFT$ (HD$,1)="M" THEN M=5
180 HD=VAL (RIGHT$ (HD$,LEN HD$-1))
190 ON M GOSUB *MON,*FIG,*CLE,*THI,*MAG
200 PRINT "........THAC0=";THACO
210 GOTO 120
220 *MON
230 PL=EVAL (HD$)-VAL (HD$)
240 HD=VAL (HD$)
250 THACO=20
260 IF PL>=3 THEN HD=HD+1:PL=0
270 IF PL>0 THEN THACO=19
280 IF HD>1 THEN THACO=21-INT (HD/2+0.5)*2
290 RETURN
300 *FIG
310 THACO=21-HD
320 RETURN
330 *CLE
340 THACO=22+INT (-HD/3)*2
350 RETURN
360 *THI
370 THACO=21+INT (-HD/2)
380 RETURN
390 *MAG
400 THACO=21+INT (-HD/3)
410 RETURN
```

次回に向けて興奮を持続させるため。

盛り上がりを高めるためにはある程度の長さを。逆に長すぎるダンジョンは2つに分割して区切りを入れましょう。

●It's Encounter!

前の部分と関連しますが，ダンジョンには敵がいるわけです。この敵もエンターテイメントの法則にしたがいます。多すぎる敵はだれるもとですが，すぐ大ボスでは盛り上がりに欠けます。

このあたりは経験がものをいいますが，慣れないうちは，固定したモンスターは2〜3ぐらいにして，ワンダリングモンスターをうまく使うといいようです。それからCRPG（コンピュータ）から入った人はモンスターを出しすぎる傾向があるので気をつけて。

●宝箱があります

そしてダンジョンには宝があるわけですが，これは資本主義の要です。しかしあまりに金を持ちすぎた奴は，利息で暮らそうとしたり地上げをしてさらなる富を得ようとします。こうなると「誰があんな危険なダンジョンに行くか！」ということになりかねません。宝物を出すのはほどほどに。

マジックアイテムも同様で，あまり強いアイテムを出してしまうと，あとはひたすらパワーゲームへと発散してしまうことが多いのです。それから，マニュアルに出ているマジックアイテムには，製作者が思いもつかなかった使い方が見つかることがあります。特に他のアイテムと組み合わせた場合，無敵のアイテムとなってしまうものがいくつか存在します。ルールの穴と呼ぶべきものですが注意しましょう。

●罠にかかった！

同様に，罠も多すぎず少なすぎずが重要です。また，罠は罠とわかってしまっては元も子もないので，害のないカムフラージュをいくつかちりばめておきましょう。

サイの目の怨念

と，今までテーブルトークRPGのコンピュータサポートについて考えてきたわけですが，世の中にはコンピュータ化してはならないことも存在します。

それは，プレイヤーのサイコロは自分で振らせなければならないのです。ひとつはインチキの防止ですが，ほかにも理由があります。

サイコロを振るということは，テーブルトークをしている過程で唯一肉体的な動作です。つまり，サイコロを振るという動作はすなわち「殴る」とか「避ける」とかいう実際の動作の代償行為なのです。

特に，ファイターなどの肉体派キャラクタは，サイコロを振る行動と攻撃とが直接対応し，サイの目を振る動作がキーとなって，プレイヤーの心理に実際に剣で殴ったのに近い緊迫感を与えるのです。

AD&DやD&Dのいいところはこのプレイヤーとキャラクタの動作の対応であり，ここに複雑化しすぎたルーンクエストや単純化しすぎたT&Tの問題点があるのだと私は思います。

ところで，サイコロを振るという動作それ自体，プレイヤーの心理的動作，すなわち意志なのですから，その結果もプレイヤーの意志力に左右されます。当然(？)サイの目は乱数にはなりません。

たとえば，私の知り合いである某後藤さんなどは確率を遥かに上回って20を出しまくります。そう，念力は確かに存在します。実はテーブルトークRPGは意志力の勝負なのです。

ついでに言っておくと，サイコロ振りにもテクニックがあります。「この角度でこうすれば1が出ない」とか「この高さからこの回転で投げれば6が出る」などというのは誰もがすることで，特に4面サイは慣れれば9割がた望みの目が出るようになります。これをインチキというか努力というかは人によってさまざまです。

その他のツール

リスト3の「monster.c」は，パソコン(＋プリンタ)用戦闘支援プログラムです。Cで書いてあるので MS-DOS 用ですがBASICで書き直すことも簡単です。Cが

テーブルトークRPGあれこれ

ゲーム名　発売元　最低必要金額(円)

Dungeons & Dragons
新和　　4,800円×2
　一番メジャー。ベーシックとエキスパートの2箱を買えばとりあえず遊べる。サプリメント（市販シナリオなど）多し。わかりやすい。

Advanced Dungeons & Dragons
TSR.inc　　15,000円ぐらい（英語版）
　米英では一番メジャー。マスターハンドブック，プレイヤーハンドブック，モンスターコンペディウム（モンスター集。現在たぶん4まで出ている）その他からなる。サプリメントもっとも多し。ルールもD&Dよりよくできている。

Rune Quest
ホビージャパン　　5,700＋7,000円
　世界設定は面白いが，なにしろ面倒くさい。

ソードワールドRPG
富士見書房　　680円
　安くていいが，サプリメントがない。

Tunnels & Trolls
社会思想社　　680円
　安いがそれだけ。

指輪物語RPG
ホビージャパン　　4,800円
　指輪の世界である。

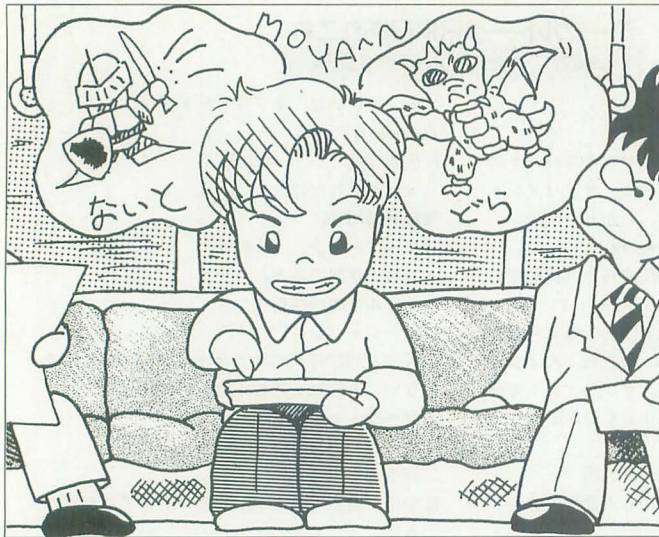
クトゥルフの呼び声
ホビージャパン
　設定が1920年代のオカルトRPG。戦闘ものではない。雰囲気を味わうためのRPG。

トラベラー
ホビージャパン　　3,500円
　SFもの。

ほかにも数多くありますがとりあえずこんなところで。
※それぞれのゲームはそれぞれ登録商標です。

テーブルトークに向いた人，向いていない人

楽しいテーブルトークといえども根本的に向いていない人がいます。どういう人かというと…
1）根本的に想像力，ロマンがない人。つまり，ゴッコ遊びができない人。
2）友達と話せない人。自閉症か，あるいは単なる嫌な奴。
3）極端な平和主義者。モンスターを殺すのが嫌いな人。

逆にテーブルトークに向いた人として，次の人たちがあげられます。
1）グループのイニシアチブを取るのが好きな人。
2）みんなとゲームをやるのが好きな人。
3）欲求不満の溜まっている人。なんでもいいから殴りたい人。
4）ひま人，または徹夜に強い人。
5）ファンタジー小説が好きな人
6）自己顕示欲の強い人
7）金持ち。
8）本を読む人，変な人。
9）英語の読める人。
（6—9は特にマスターをやる人）

ある人は，とりあえず「monster.c」というファイル名で入力して，コンパイルしてください。実行は，

monster hd n

とします。hdは，モンスターのHDで，nは数です。そうすると，標準出力にそのモンスターのHPがn個分計算されて出てきますから，プリンタにリダイレクトすればたくさんモンスターが出たときにいちいちHPを振らないでもよくなります。また，よく使うモンスターのHPは作り置きしておくと便利です。

また，長くなるのでリストは載せませんが，名前生成プログラムというのもありまして，暗号関係の本に英語の2文字連関表（たとえばTの次にHのくる確率は何%で…という表）というものがあるのでこれと乱数を使えばなんとなく英語っぽい単語をたくさん作ることができます。これもプリンタに落としておくと名前がたくさん必要なときに役に立ちます。

皆さんがんばってね

というわけで，なんとなくこのコーナーは終わりですが，私としてはもっとテーブルトークがメジャーになってルールブックやサプリメントの値段が下がることと，女性RPG人口が増えることを（その点ではOh!Xに連載することはなにか間違っているような気がする）願ってやみません。どこかのネットでAD&D, D&DのSIGを開いてくれるといいんですけれどねぇ。

それでは皆さん，ごきげんよう。

《RPG参考図書》

指輪物語　（全6巻）　評論社文庫

言わずと知れた指輪である。文章はかなりゲボボだけれど一般教養ということで。

ゲド戦記　（全3巻）　岩波書店

1巻だけでいいような気もするけどまあ面白い。

リフトウォーサーガ（魔術師の帝国，シルバーソーン，セサノンの暗黒）　早川文庫FT

まるでRPGのシナリオのよう。物語としてはまあまあ。

ベルガリアード物語　（全5巻）　早川文庫FT

これもRPGを見ているような話。続編も出てなんとなく銀英伝化している。展開はたるいが結構面白い。

ハロルドシェイ　（全4巻）　早川文庫FT

現代人がファンタジー世界に入るという，最近ありがちな小説のたぶん元祖である。

霜のなかの顔　早川文庫FT

魔法使いもののなかではかなりの出来。

ドラゴンランス戦記　富士見書房

AD&Dのシナリオを本にしたもので，続編，外伝などたくさん出ている。そんなに面白いわけではないが，できればペーパーバックで読むと英語の練習になる。アメリカ人はテーブルトークで恥ずかしげもなくよくこんなセリフをはけるもんだと思う。

その他，ギリシャ・ローマ神話，北欧神話など

要約版も出ているのでとりあえず読んでおいて損はない。

リスト3 monster.c

```
 1: /*
 2:  *      monster
 3:  *
 4:  *      exec..  monster hd n
 5:  */
 6:
 7: #include <stdio.h>
 8: #include <math.h>
 9: #include <string.h>
10: char buf[50];
11:
12: main(argc,argv)
13: int argc;
14: char **argv;
15: {
16:         int n,j;
17:         int hp;
18:         int hd;
19:         int plus=0;
20:
21:         if (argc==3) {
22:                 strcpy(buf,*++argv);
23:         }
24:         else {
25:                 fprintf(stderr,"HD=");
26:                 gets(buf);
27:         }
28:         printf("***** HD=%s *****¥n",buf);
29:         plus=0;
30:         for (j=0; buf[j]!='¥0'; j++) {
31:                 if (buf[j]=='+') {
32:                         buf[j]='¥0';
33:                         plus=atoi(buf+j+1);
34:                         break;
35:                 }
36:                 if (buf[j]=='-') {
37:                         buf[j]='¥0';
38:                         plus=-atoi(buf+j+1);
39:                         break;
40:                 }
41:         }
42:         hd=atoi(buf);
43:         if (argc==3) {
44:                 n=atoi(*++argv);
45:         }
46:         else {
47:                 fprintf(stderr,"N=");
48:                 gets(buf);
49:                 n=atoi(buf);
50:         }
51:         while (kbhit()) getch();
52:         fprintf(stderr,"Hit any key!¥n");
53:         for (j=0;;j++) {
54:                 if (kbhit()) break;
55:         }
56:         getch();
57:         srand(j);
58:
59:         while (n--) {
60:                 hp=0;
61:                 for (j=0; j<hd; j++) {
62:                         hp+=rand()%8+1;
63:                 }
64:                 if ((hp+=plus)<1) hp=1;
65:                 printf("%4d        ",hp);
66:         }
67:         printf("¥n");
68: }
69:
70:
71:
72:
```

●特集

電子音楽術入門

コンピュータミュージックといえば，いまや誰もがMIDIを思い浮かべることでしょう。パソコンの内蔵音源（FM音源）に注目する人は少ないかもしれません。しかし，内蔵音源の性能は少々低く見積もられ過ぎではないでしょうか。
実際，X68000に標準装備のミュージックドライバが内蔵音源の性能を生かしきるものでないのも事実です。しかし，使い手によってはYコマンドの多用という特殊な方法によって高度な表現を実現しています。また，その手法をプログラムで自動化する試みもあります。ここでは，よりよい音を作るための知識，自由な表現のためのテクニックを紹介していくことにしましょう。
なにより，ユーザーがパソコンを通じて音に触れ，曲を楽しみ，作品を生み出し，そして音楽的素養を身につけていく……，そんなプロセスが考えられるとしたら，それこそがパソコンの目指すべき音楽環境の姿ではないでしょうか。

CONTENTS

PIXYとX68000とMML

FM音源のある部屋

Ogikubo Kei 荻窪 圭

FM音源はゲームのBGMのためだけにあるわけじゃない。朝起きたときも，プログラミングをするときも，そして寝るときも，FM音源があれば，あなたは音楽とともに暮らしていけるのだ。さあ，君もそんな自分だけの部屋を作ろう。

いやあ，実は先日，PIXYを買ってしまいまして。あの,「PIXYの幅があれば生きていける」のPIXYでんがな。しかも，いちばん高いPIXY99X。どうして99Xかというと，DSPを搭載している！ てのに惹かれてしまったのだ。しかも，外部入力がVIDEO1～3の3系統ついていて，AVセレクタにもなるという。VIDEO1はビデオで，2はX68000で，3はアナログプレイヤーという構成。ほかの機種はAVセレクタを買わないと端子が足りなかったのだ。しかも，映像IN/OUTもある。

X68000のFM音源の音というのは，Yコマンドとか使って器用なことをすればかなりのところまでいくが，イマイチ，音に厚みがない。そこで，DSP付きのPIXYならばいろいろエフェクトできるのではないか，と思ったわけだ。

PIXY99Xにはダイナミックサウンド，パラメトリックイコライザー，サラウンドの3種をDSPで行っている。最初のは，コンプレスを行って，小さい音を強める効果を出すもの。イコライザーはグラフィックイコライザー。サラウンドはお馴染みサラウンドで，残響効果のレベルと残響時間を設定できる。せっかくDSP積んだのだから，もっといろんなことができてもいいと思うし，もっと細かい設定もしたいなあ。ついでにRS-232Cなんかがあって，DSPの効果をX68000から制御できたら面白いだろうになあ。でもそれ以上いうときっと定価がぽんと上がったりしそうだから，泣く泣く妥協というところだ。

で，テープなんかに落とすときもDSPをかけてから落とせるので，X68000の音を派手にして録音なんてのも可能だ。

DSPは知ってのとおり，デジタルシグナルプロセッサであり，デジタル信号のリアルタイム処理に向いている強力なプロセッサだ。CDのサンプリングレート約44kHzくらいなら，安いチップで十分だろう。

CD以外のソースについては，きっと，A/D変換してDSPを通して，D/A変換してアンプへ渡していると思う。

それで，だな。どうしてPIXYの話をしたかというと，こいつがまたX68000とぴったりあうのだよ。推奨オーディオシステムといってもいいくらいだ。

なぜなら，PIXY99Xを全部積み重ねたときの高さが，X68000と1㎜程度しか違わないのだ。そんでもって，PIXYはちょっとざらっとした黒なので，X68000のチタンブラックとよくマッチするのだ。私のは初代だからグレーと黒という取り合わせになってしまっているが，ブラックのX68000なら，もう最初からX68000の周辺機器として作ったのかと思わせるほど。マンハッタンシェイプと高さが一緒というのは驚いたな。

さあ，ここまで褒めたからにはSONYから何かもらいたいところだが（ハンディカム45がいいな)，そんなに甘くないのが世の常である。

そして私は昔作ったOh!Xのテーマなんかを DSPでサラウンドして聴いて感動していたりするのであった。バッハなんて派手にサラウンドすると教会の残響の雰囲気がちょっとだけ出て，気持ちいいぞ。

でも，ビープ音がサラウンドするのはとっても気持ち悪い。

MML文化

で，日本にはアメリカにはない独自のパソコン文化がいくつかある。ひとつはアニメ絵文化であり，もうひとつがMML文化だ。どちらもサブカルチャーって感じだ。

MMLっていうのは知らない人がいてもおかしくないのでいっておくと，ミュージックマクロランゲージの略で，ランゲージというからには言語なのである。ふつう，このMMLってので書いた演奏データをFM音源ドライバに送ってやると，FM音源が勝手に音を出す。

このMML文化の面白いところは，OS自体はMMLをFM音源のドライバがそのまま受け付けられるデータに翻訳して，FM音源に渡してやるだけっていうこと。あとは，ユーザーはなにもしなくていいのである。だから，演奏が始まれば，あとは，ワープロしようがプログラム作ろうが絵を描こうがなにしていてもいいのだ。FM音源は勝手に，もらったデータを演奏し終えるまで鳴っているのである。

確かにFM音源はたった1個のFM音源にすぎない。MIDIで制御する最新の音源に比べれば音も貧弱だし，アンプまわりの回路もオーディオ機器に比べればよくもない。でもだなあ，だからってFM音源を馬鹿にしたりゲームのBGM専用だと思うのだとしたら，それは1600万色でないからX68000のCGは貧弱だとか，1500dpi以下の印刷物なんて読む気がしないっていっているのと一緒だ。パソコンが内蔵している程度のたった8チャンネルの音源だって，観賞用としてはCDやMIDI楽器なんかには及ぶべくもないけど，積極的に楽しめばとても面白いものなのだ。PIXYのない人も，ここはひとつFM音源のある部屋を作ってみようではないか。

●FM音源のある部屋1

朝，机の下で寝ている猫だけが気づく小さな音でクィンとディスプレイの電源が入り，ハードディスクが回り始める。それに呼応するように黒一色のオーディオシステムも立ち上がり，じっと入力を待つ。

ハードディスクからたっぷりとプログラムを吸い取ったX68000は息を始め，AUTOEXEC.BATに仕込まれた小さなプログラムが今日の日付と時間をチェックして予定された目覚めの1曲を選ぶ。

突如として大音量で鳴り響くバッハのト短調フーガが30秒くらい部屋を満たしたのち彼は目を覚まし，枕もとのリモコンで音量を適正ボリュームまで下げる。

バッハを鳴らしながらCRTに広がる今日の時間割りを呆けた目で睨み，OPT.2+.でディスプレイを朝のニュースに変更する。台風が来ている。今日は学校を休もう。今

月2回目の風邪だがいいや。熱が39度くらいあることにしよう。

解説：バッハの曲は，特にオルガン用のカノンやフーガなんかは相手がオルガンであるからして，弦楽器のように微妙な表現ができない。そして，ある意味で数学的である。8チャンネルのFM音源でも工夫すればかなりの音が表現できる。

朝，X68000が立ち上がるのはタイマー機能のおかげである。毎日違う曲が鳴るのは，その日の日付（あるいは曜日）をチェックするプログラムを用意したからである。チェックした結果はエラーコードという形で数字となって返ってくる。バッチプログラム内の"IF ERROR LEVEL"文で返ってきた数字に応じたラベルへ飛ばす。そこには"copy BACH_Gm.OPM opm"って書いてあるだけだったりする。

この毎日違う曲というのがミソで，毎回同じ曲が鳴っていたら，いくらなんでも飽きる。

● FM音源のある部屋2

どたどた歩くと階下から苦情の出るフローリングの部屋の真ん中にリラクゼイションチェアがある。ボディソニック付きだ。それにゆったりと座る。視線の先には21インチのディスプレイがあり，X68000がつながっている。X68000の音声出力は20素子のイコライザとデジタルリバーブを通してリラクゼイションチェアにつながり，そこから伸びたケーブルには高価なヘッドホンがある。

指をキーボードに伸ばしてFM音源の曲を連続プレイでいくつかセットする。静かでかつ刺激的な環境音楽が鳴り始め，それに同期をとったサイケデリックなパターンが21インチディスプレイをうごめく。

やがて，サイケデリックなパターンが彼を包み込み，異世界を垣間見たようなサイバーなトリップに旅立つ。

解説：FM音源と同期をとってなにかを行う試みは，Oh!X 1989年6月号のKENBAN.BASで行われている。その後も，チャンネル別レベルメーターなど，演奏中画面を飾る環境ソフトはいくつもある。AMIGAにはマイクの拾った音にあわせて画面に画像パターンを表示するというハード＆ソフト（マインドライト7）が発売されている。X68000でもFM音源の演奏にあわせてサイケデリックなパターンを表示させるソフトは作成可能である。

サイケデリックという言葉には「ドラッグが，日常に垢で汚れ，曇った心を洗い流してくれるという発想がある」(STUDIO VOICE 9月号より)。どんなパターンを描くかについては，いろいろ歴史的に研究された資料がある。たとえば，荘厳にはすみれ色，興奮には赤か赤紫。音程と色との対応。一部は「色を心で視る」という本に紹介されている。

私は常々，こういった環境ソフトを待ち望んでいるが，作ろうという人はなかなか現れないようだ。リアルタイムで音符を出したりレベルメータしたり，っていう発想が出つくしたら，ぜひこういった遊びにも挑戦してもらいたいものだ。

● FM音源のある部屋3

彼はたくさんのFM音源を演奏するプログラムやOPMファイルを持っているが，彼のプログラムには特徴がある。チャンネル1から6までしか使わないのだ。

あとの2つはなにかというと，アドリブ用である。チャンネル7と8を使って，彼はアドリブでシンバルなり効果音なりを入れる。

彼は夜中，退屈してくるとX68000とオーディオユニットに火を入れ，その日の気分で音楽を鳴らす。X68000の音を無機質にポルタメントもなにもかけないで硬い音で鳴らすと気持ちいいとのたまうレトロテクノボーイだ。ただ聴いているだけだと退屈なので，アドリブでちゃちゃを入れる。ちょっとだけ，演奏に参加した気分になる。

解説：X68000では演奏データをトラックというバッファに入れ，そのデータをチャンネルごとに演奏させることができる。使っていないチャンネルはいつでも使用可能である。アドリブだって，できる。

［BASICサンプル］

```
10 for I=1 to 8
20   m_alloc(I,1000)
30 next
40 m_trk(1,"演奏データ")
50 m_trk(2,"演奏データ")
        .
        .
100 m_tempo(120):m_play(1,2,3,4,5,6)
110 str S
120 while -1
130   I=INKEY$
140   if I=" " then m_play(7) else m_play(8)
150 endwhile
```

そろそろネタが虚しくなってきたのでこのシリーズはやめよう。

まずは聴くことから始めよう

なんだかんだいって，自分で演奏データを作るのはむずかしい。でも，みんな，なにがしかのデータを持っているはずだ。

X68000においてFM音源を鳴らすには2つの方法がある。ひとつはOSの管理下でOPMファイルと呼ばれるものを鳴らす方法であり，もうひとつがX-BASIC上でMMLを使ったプログラムで鳴らす方法である。

OPMファイルといえば，SX-WINDOWを買った人ならサンプルがついてきたはずだ。拡張子が「.OPM」のファイルである。電脳倶楽部購読者なら毎月OPMファイルがついてくる。電脳倶楽部に載っている音楽データはすべてOPMファイルだ。

OPMファイルというのはOPMドライバ(FM音源ドライバで，OPMDRV.Xという名前）で扱える形式の演奏データをいう。その形式のデータを，

copy ファイル名 opm

とOPMにコピーしてやれば，演奏してくれるのだ。このいろんなデバイスをファイルと同等に扱うというのはUNIX以来の伝統を引き継いでおり，X68000のおいしい特徴のひとつだ。PRNにコピーするとプリンタに，AUXにコピーするとRS-232Cに，PCMにコピーするとADPCMに，CONにコピーするとコンソール(つまり出力ならCRT，入力ならキーボードで，標準入出力という）に出力されるのもそうだ。

続いて，BASICのMMLだが，これはOh!X LIVEやシステムディスクについてくる演奏プログラムなんかがそれにあたる。BASICがMMLを解釈して，OPMにほらよ！　と渡してやる方法だ。

OPMファイルに関しては次の記事で，X-BASICのMMLに関しては以降の記事を参照のこと。

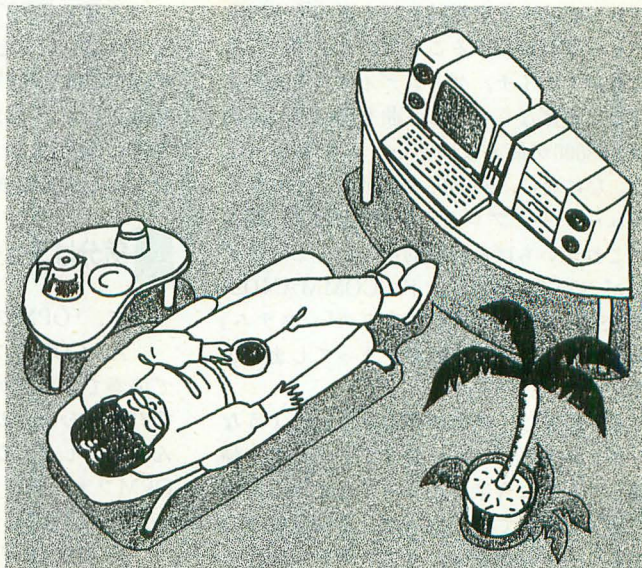
それでもって，ステレオでFM音源を聴いてみれば，そこそこの音が出ることに気づくはずだ。

まだの人は早くFM音源のある部屋を作ろう。

参考文献

[1] STUDIO VOICE'90年9月号（特集 ACID AGE)
[2] 色を心で視る，福村出版

参考体験

[1] ブレインマインドジムPSYのシンクロエナジャイザー
[2] SONY PIXY99XのDSP

システム上でBGMを

OPMファイルで遊ぼう

Nishikawa Zenji 西川 善司

OPMファイルは Human68 上のミュージックドライバOPMDRV.Xがサポートする音楽データの標準形式です。マニュアルなどには載っていませんが、本誌でもしばしば登場しますので、これを機に扱い方を覚えておいてください。

いやぁ，この間，某駅前通りを歩いていたときやられてしまいましたよ。突然，背後から「ちょっと時間をいただけますか」ときて，返事をする前に「この中から興味のあるものを選んでください」ときました。若い女性でした。「こ，この暑い中，なんてご苦労なヤツなんだ」と思いつつ選択肢を覗き込みむと，そ，そこには……，

「結婚，恋愛，宗教，超能力，死後の世界，異性，戦争，天皇，……，科学」

しばしの沈黙ののち私は「…か，科学…」と答えると，彼女は「あ，皆さんそう答えるんですよね」と。「だってあんた，なにが悲しくて町の真ん中で見ず知らずのあんたと死後の世界を語り合わなきゃならないんだよ」と思いつつもあとの質問に全部答えてしまう人のいい西川善司でした。

OPMDRV.Xがあるじゃないか

皆さんはX68000でどのように音楽を楽しんでいるのでしょうか。え？　ああ，なるほど，市販ソフトのミュージックモードを使って聞いている，と。それは，なかなか「通」ですな。私もよく「ちぃ，暑いときにゃ，これにつきるぜぇ」といってドラゴンスピリットを起動し，１面のテーマを流しっぱなしにすることがあります。でも，これはイチイチ，ゲームディスクを立ち上げなければならないし，曲を聴いている間はX68000が使えないというのがちょっとね。なになに，BASICを起動してそれからミュージックプログラムをRUN？　確かにこれがいちばん一般的なのかもしれませんが，SYSTEMコマンドでCOMMAND.XやVS.X, SX-WINDOWなどの「システム」に帰還したときには曲が止まってしまいます。

多くの人は，仕事や作業を上記のようなシステム上で行っているはずですから，帰還したときに音楽が止まってしまうというのは少し残念です。やっぱり音楽を聴きながらワープロを使ったりプログラムを作りたいものですよね。

では，本体同梱のFM音源ドライバ「OPMDRV.X」はこういったシステム上で音楽を鳴らすようにはできていないのでしょうか？　いえいえ，使いやすさにこだわるシャープがそんな陳腐なものを作るはずがありません。

BASICの外部関数にMUSIC.FNCというのがありますがこれは実は関数が受け取ったパラメータをこのOPMDRV.Xへ転送しているにすぎません。ですからシステム上からOPMDRV.Xへパラメータを送ることができればシステム上でも音楽を鳴らすことはできるわけです。この送るべきパラメータをファイルにしてしまったのが通称「OPMファイル」です。

OPMファイルは通常の文書ファイルと構造上はまったく同じもので，アセンブラやＣのソースプログラム同様，エディタ(ED.X) などで１行１行にOPMDRV.Xのコマンドやパラメータを記述して作成します(行番号はいりません)。できたOPMファイルはシステム予約ファイル「OPM」へCOPYすればOPMDRV.Xへデータを渡してくれます。いまTEST.OPMというファイルがあったとすれば，

```
COPY TEST.OPM OPM
```

とすればいいことになります。しかし，必ずOPMDRV.Xがデバイスドライバとして登録されていなければなりません。でないと単にOPMというファイルが作られるだけで，なにも起きません。

気分はミュージックモード!!

さて，「OPMファイル」は，

```
A>COPY ファイル名 OPM
```

で演奏できますが，どうもファイルネームを打ち込むあたりがスマートじゃありませんね。少しましな方法があるにはあります。OPMファイルのファイルネームを「ファイルネーム.OPM」で統一し，エディタで，

```
ECHO OFF
COPY %1.OPM OPM > NUL
```

というようなバッチファイルを「PLAY.BAT」といった名前で作るのです(バッチファイルを知らない人は本誌１月号のOS特集やHuman68kマニュアルを読んでチョンマゲ)。

これで，

```
A>PLAY ファイルネーム
```

(「.OPM」まで打ち込む必要はない)で演奏ができるようになります。

うーん，だいぶキータイプ数は減っているものの，まだファイルネームを打ち込むあたりが面倒ですね。ここで待ってましたとばかりに登場してくるのがビジュアルシェル「VS.X」。このVS.Xのマウスオペレーションによるメニュー機能を利用して市販ソフト顔負けのミュージックモードを作ってしまいましょう。

用意するものはファイルネームの拡張子を「.OPM」で統一したOPMファイルとVS.Xだけです。それでは，まず，VS.Xを起動してください。VS.Xの画面右側中央下あたりに机の上に本やら三角定規が置いてあるアイコンがありますね。これをマウスカーソルでポイントし，右ボタンを押しながら「Icon Maintenance」を選択してください。

ここで音符など，なにか音楽に関係する絵を適当に描いたら，画面上の「アイコン名」というところをクリックして

```
*.OPM
```

と打ち込んでリターンしてください。拡張子が「.OPM」のファイルすべてがこのアイコンの絵で表示されます。次に「実行ファイル」のところをクリックして，

```
/COMMAND.X
```

と入力(「/」を指定しないと実行終了後に「vs.x:press mouse button or key」と出てきてしまいちょっとみっともない)，さらに「パラメータ」をクリックして，

```
COPY % OPM
```

と打ち込んで，「OK」で「Icon Maintenance」を終了します。これでひとたびこの

アイコンがクリックされると，

COMMAND.X COPY % OPM

が実行されるわけです。これは，子プロセスでCOMMAND.Xを起動し，その直後に「COPY % OPM」を実行したのと同じ意味になります。「%」にはアイコンのファイル名が代入されるので実質，

COPY OPMファイル名 OPM

が実行されるわけです。

これで「音楽ビジュアルシェル」の完成です。好きな曲を選択したら，ワープロでも起動して「本邦初！ BGM付きワープロ！」とかいってPCなんとかの一〇郎ユーザーにでも自慢してください(「なんだそりゃ」と返り討ちにあうこと請け合い!?)。

ところで，SX-WINDOWを持っているユーザーはこのようなことをしなくてもOPMファイルを「アクセサリ」の「プレイヤー」アイコンへ持っていくだけで簡単に演奏ができてしまうんですよね。

*　　　*

メーカーから与えられたシステムディスクのみで，これだけの音楽環境を整えることができるパソコンってもしかしたら「X68000」だけかもしれませんね。いやぁ，日本に生まれてよかった，X68000ユーザーでよかった。

OPMファイルの文法

OPMファイルを作成するには「OPMファイルフォーマット」を知る必要がありますが，BASICの「音楽用外部関数」を理解していればなにも難しいことはありません。ここではBASICの音楽外部関数と対比させてOPMファイルフォーマットを説明していくことにしましょう（音楽外部関数についての詳しいことはBASICマニュアルを参照してください）。

```
m_alloc(TR,SZ)
(mTR,SZ)
```

m_allocはMML用のトラックバッファを確保する命令です。トラック番号TRにSZバイト確保しますが，これをOPMファイルフォーマットでは上記のように括弧で閉じた中に「m」と2つのパラメータをm_alloc命令と同じ順番で記述します。

例)

m_alloc(4,5000)　は，(m4,5000)

となります。

```
m_assign(CH,TR)
(aCH,TR)
```

m_assignはm_allocで確保したトラックバッファをFM音源のチャンネル番号CH（1～8）に割り当てます。ですからTRはm_allocのパラメータのTRと同じものです。OPMファイルでは上記のように括弧で閉じた中に2つのパラメータをm_assignと同じ順番で記述します。

例)

m_assign(1,4)　は，(a1,4)

となります。

```
m_vset(I,V)
(vI,0,……)
```

m_vsetは4×10の音色パラメータ配列変数Vを音色番号Iへセットする命令です。OPMファイルフォーマットではまず「(」の後ろに音色セットのコマンド「v」を書き，その後ろにダミーのゼロを書きます。あとは配列Vの内容を，

V(0,0), V(0,1), V(0,2), …, V(0,10), V(1,0), …, V(1,10), …, V(2,0), …

と順番に55個の数値を「,」で区切りながら書き，最後に「)」でしめくくります。

(BASICで表すとしたら，

```
FOR I=0 TO 4：FOR J=0 TO 10：
PRINT V(I,J)
:NEXT:NEXT
```

という感じかな)。

具体的な例を図に示します。

```
m_init( )
(i)
```

m_initはFM音源やOPMDRV.Xのイニシャライズを行うコマンドです。OPMファイルフォーマットでは「(i)」の3文字を書くだけです。

```
m_tempo(T)
(oT)
```

m_tempoは読んで字のごとく，テンポを設定するコマンドです。OPMファイルフォーマットでは，テンポの値Tをコマンド文字「o」の後ろに書きこれを括弧でくくります。

例)

m_tempo(156)　は，(o156)

に相当します。

```
m_trk(TR,MML)
(tTR)MML
```

m_trkは実際にMMLをOPMDRV.Xにセットする命令です。OPMファイルフォーマットでは，括弧でくくられた中にコマンド文字「t」とトラック番号TRを書き，閉じ括弧「)」の後ろにずらーっとMMLデータを書きます。もちろん，MMLデータは通常BASICで使用しているのとまったく同じものでかまいません。ところで，OPMファイルフォーマットではMML文字列の長さは255文字以内という制約がありませんので，一度も改行せず，力まかせに書くことも可能です。

例)

m_trk(1,"cdefg")　は，(t1)cdefg

と書きます。

```
m_play(CH1,CH2,…,CH8)
m_stop(CH1,CH2,…,CH8)
m_cont(CH1,CH2,…,CH8)
(pCH1,CH2,…,CH8)
(sCH1,CH2,…,CH8)
(cCH1,CH2,…,CH8)
```

いずれも演奏を直接制御するコマンドですがとても似かよっているのでまとめて説明します。m_play，m_stop，m_contはそれぞれ「演奏開始」「演奏中断」「演奏再開」のコマンドで，引数にFM音源のチャンネル番号をとることによって指定されたチャンネルへの制御が可能です。

たとえば，

m_play(1,2)

はFM音源のチャンネル1と2のみの演奏を開始します。また，この引数を省略すると全チャンネル対象の制御となります。つまり，

m_play( )

は，

m_play(1,2,3,4,5,6,7,8)

に相当し，全チャンネルの演奏を開始します。

さて，OPMファイルフォーマットのコマンドにもこれらに相当する命令があるのはもちろんですが，引数も同様に省略したり設定したりすることができます。

m_play(1,2)　は，(p1,2)　に，

m_play( )　は，(p)

に相当します。いまの例の「p」を「s」に変えるとm_stopに，「c」に変えるとm_contの機能をします。

*

OPMファイルフォーマットのコマンド説明はこれで終わりです。しかし「エディタでミュージックプログラムを書くのはどうもねぇ。変数やfor～nextといったループ制御文も使えないし」といまいちOPMファイルに魅力を感じない人もいることでしょう。確かにミュージックプログラムの開発環境についてはBASICのほうに軍配が上がります。

そこで，今回はBASIC上で作ったミュージックプログラムをOPMファイルに変換するプログラムを特集の最後に掲載しています。ぜひとも利用してください。

図1 m_vset

「BASIC」で

```
140 /* AF  OM  WF  SY SPD PMD AMD PMS AMS PAN
150 v={56, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
160 /* AR  DR  SR  RR  SL  OL  KS  ML DT1 DT2 AME
170    26,  8,  5,  7,  2, 30,  3,  3,  3,  0,  0,
180    29,  5,  4,  4,  1, 30,  3,  4,  3,  0,  0,
190    28,  4,  2,  6,  3, 30,  3,  1,  3,  0,  0,
200    31, 10,  3,  5,  1,  0,  3,  1,  3,  0,  0}
210 m_vset(70,v)
```

「OPMファイル」では

```
(v70,0,56,15,0,0,0,0,0,0,0,3,0,26,8,5,7,2,30,3,3,3,0,0,29,5,4,4,1,30,3,4,3,0,
0,28,4,2,6,3,30,3,1,3,0,0,31,10,3,5,1,0,3,1,3,0,0)
```

こうする。

初心者から中級者に贈る

音楽プログラムは怖くない

Nishikawa Zenji 西川 善司

「音楽プログラム」というものが市民権を得て以来，数多くのプログラムが雑誌に掲載されている。が，それでも音楽データの入力に抵抗を感じる人は多いようだ。ここではMMLプログラミングの初歩的な事項を確認してみよう。

私が[CTRL]+[OPT1]+[DEL]でリセットをかけるのが癖となりハードディスクアクセス中にこれをやって重要なファイル(実はOPMD.Xのソースリスト)を失ってしまったマヌーな西川善司です。ここでは，

・いかに楽しく音楽プログラムを入力するか
・いかに楽をして音楽プログラムを作るか
・いかに掲載されるような音楽プログラムを作るか

などをお話ししようかと思います。「なにいってやんでい，この馬鹿は！」と感じる部分もあるかもしれませんが，そのときはご容赦くださいネ（と先にわびを入れる腰の低い善司）。

X68000の内蔵音源はFM音源８声とAD PCM1声の９重和音です。Oh!Xでは，従来のデータとの互換性を保ちつつもFM音源とAD PCMを同期する「OPMA」ドライバの発表以後，この２つの音源を駆使した素晴らしい音楽プログラムが数多く発表されてきました。もういまでは市販ゲームソフトのBGMもこのFM＋AD PCMの同期演奏は常識（ソフトハウスには悪いけど）いい風潮になったもんです。

と，そんなわけで，せっかくこれほど音楽機能の充実したパソコンを所有しているんです，他人の音楽プログラムを入力するのも結構，自分で譜面を打ち込んでみるのも結構，とにかく自分の手で直接音源をいじってみませんか。

入力はつらくない

葉書にこんなのがありました。

「私はダンプリストは何Kバイトでも入力できるが，ミュージックデータを見ると鳥肌が立つ」　山口県　出留門手斗磨夫

うーん。私はあの「さススっ」という黒板消しが黒板と擦れる音を聞くと鳥肌が立ちますがミュージックデータを見て鳥肌が立つという人は初めてですな。出留門手さんは恐らく入力のコツをまだ知らないのかもしれませんね。

ところで，機械語のダンプリストというのは入力の途中で実行することはまずできませんが，ミュージックプログラムはそれが可能です。つまり，入力した時点までを聴くことができるのです。

たとえば，いま主流と思われる，プログラムを**８つのブロックに分けて**そのブロックブロックが各チャンネルに対応しているタイプがあります（今月の「ZMUSIC.FNC」のサンプルプログラムなどがそうです。以下の文章はそのようなリストを見ながら読むとわかりやすいと思います）。こういったタイプは頭からただ順番に打ち込むのではなく，プログラム先頭の音色設定部分や文字変数宣言の箇所まで打ち込んだら８つに分かれた各ブロックを並行して入力するようにするのです。

音色設定部や変数宣言部を入力後，たとえば各ブロックの「a=」という文字列とそれをトラックバッファにセットしている命令（つまり「m_trk(?,a)」ですね）を入力します。それから「RUN」するとエラーがなければ「OK」と表示されますので，

m_play ( )

を実行してください。入力し終えたところまで曲を聴くことができます。

こうして同様に文字列を入力していくたびに試聴していけば，**入力ミスを発見しやすくなるし**，そのまた次の文字列を入力したくなってきます。ちなみに音楽プログラムを作る側も多くの場合こうした，ちょっと打ち込んでは試聴……という方法をとっていますよ。

ところで６月号付録ディスクに収録されているMUSICDRVのような高機能MMLは別にして，普通MMLでは「１トラックでは１声しかシーケンスできない」という鉄則があります。したがって主に和音で構成されるコードやバッキングといったパートは複数のトラックに分散されてプログラムされています。しかし，互いのトラックが非常に似通った文字列になっていることが多いのでこうしたトラックは入力した文字列を「LIST」コマンドなどで再び画面に出しこれをエディットすると非常に楽です。

たとえば図１のような譜面があったとします。この和音のルート「C」がMMLになっていたとすると，

L16C8.CC4C8.CC4

ですが，次の「E」を入力するときは上の文字列の「C」の上にカーソルを合わせて「E」を入れていけばよいのです（音長のMMLはいじらない）。万が一，音長部分の入力ミスをしてしまった場合はあとで試聴したときに和音まるごとテンポがおかしくなってくるはずなのですぐわかります。

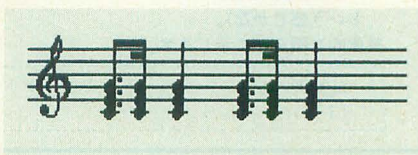
見やすいミュージックプログラム

BASICマニュアルの「m_trk」の項を読んで基本的なMMLを覚えたら，多くの人が楽譜を基にミュージックプログラムを作ると思います。譜面を基に作る場合，いくつか注意したいことがあるのでここではそれについてお話しします。

MMLは一般的なスコア入力ソフトと違ってとっつきにくいものですが，ひとたびこれに慣れてしまうとほかのミュージックソフトとは比べものにならないほど高速に１曲を作り上げることができます。

しかし，どうも演奏がおかしいと感じていざプログラムを修正しようと思ったときには，譜面のどの部分がプログラムリストのどこに対応しているのかわからなくなってしまうことが多いのです。そんなわけで，なるべく美しい，見やすいミュージックプログラムを作ることを心がけなければならないわけですが，

a）１小節入力したらスペースを開ける

図１

b) コメントを入れる
c) ひとつの文字変数に何小節もつっ込まない
d) 「L」は大文字で書く

の4点に気をつければそこそこのものが書けるはずです。

a)は演奏が各トラックバラバラになってしまうといったバグを発見するときに便利です。こうしたバグの原因はほとんどの場合，音長関係の入力ミスですのでMMLと譜面に間違いがないかどうかいちいち見ていくしかないように思われますが，a)のように1小節ごとに区切っておけば，譜面のほうを見る必要はありません。というのは打ち込んだMMLがちゃんと譜面の拍子にあっているかどうかをチェックしていけばいいからです。

たとえば4分の4拍子の場合だと，音長関係の入力ミスをしてしまった場合は必ずどこかに(その小節内に存在する音符の)音長をすべて足したときの和が4分音符4つ分になっていない小節があるはずです(譜面が間違っている場合もありますから注意しましょう)。私の経験上，「付号」音符が多い小節でこうしたミスが起こりやすいようです。

b)はプログラムを組んでしばらく時間をおいて見直したときにわかりやすくするものです。たとえば「この文字列に代入されているMMLはいったい譜面のどの部分なんだ？」と思ったときに「繰り返し2回目」とか「ピアノソロ」など譜面情報や楽器情報を書いておけば容易に譜面との対応が把握できます。

c),d)は純粋にプリントアウトしたときや，「LIST」で画面に出力したときに見苦しくしないようにしたり，誤読を防ぐためのものです。

超初心者向けの音色作成法

ゲームミュージックやポップス，フュージョンにしろ，作っている曲をなるべく原曲に似せようとした場合，まず問題となるのが音色です。音色をまったくゼロの状態から作り始めるのはとても難しいことなので内蔵音色などから似た音を持ってきてそれをエディットして目的の音色へ作り変えるのが一般的です（リスト1に内蔵音色を配列変数にコンバートするプログラムを載せておきます)。

まず似た音色の選び方ですが音量変化が似ている音よりも音質が似ているものを選んだほうがよいでしょう。また，音色名にこだわらないで選び出すことも大切です。希望している音色がまったく関係のなさそうな楽器名で登録されているかもしれませんよ。

さて，エディットするにはFM音源についての知識をある程度持っていなくてはなりません。

X68000のFM音源は4オペレータ・8アルゴリズムと呼ばれるタイプです。4オペレータとは音を構成する要素が4つあり，このオペレータのつなぎあわせ方「アルゴリズム」が8通りあるという意味です（図2)。また，オペレータにはモジュレータとキャリアの2種類があり図2では網掛けになっているのがキャリア，そうでないものがモジュレータです。

キャリアというのは図2を見るとわかりますがどのアルゴリズムの場合でもいちばん下にきているものですね。難しいことは専門書に譲るとして，初心者は，キャリアとは「音量変化を決めるものである」，モジュレータは「音質変化を決めるものである」と覚えましょう（本当はちょっと違うけど)。

ところで，すでにある程度似た音を選んできている場合はこの音質変化というのはあまり重要ではありません。では目的としている音色に作り変えるにはどこをエディットしたらよいのでしょうか。それは，トータルレベル（TLまたはOLと略される。以下括弧の中は略号とする)」と「マルチプル(ML)」に注目すればよいのです。この2つのパラメータの意味については，またまた専門書に譲りますが音質そのものを決めるパラメータだと思ってください。これらをいろいろいじってみて目的の音色にいちばん近い状態にします。イメージ的なアドバイスを書いておくと，

リスト1 音色のファイル化

```
 10 char st=1,ed=1,g
 20 int i,j,k,ln=100,fn
 30 dim char v(4,10)
 40 str s[256]
 50 input "from";st
 60 input "to  ";ed
 70 input "line number";ln
 80 input "make file ? (y/n):",s
 90 if s="y" or s="Y" then mkfl():g=1 else g=0
100 for k=st to ed
110    m_vget(k,v)
120    s=right$("     "+str$(ln),5)
130    prt(s+" /*  AF    OM    WF    SY    SP  PMD  AMD  PMS  AMS  PAN")
140    ln=ln+10
150    s=right$("     "+str$(ln),5)+" v={"
160    for i=0 to 10
170       s=s+right$("   "+str$(v(0,i)),3)+", "
180    next
190    prt(s)
200    ln=ln+10
210    s=right$("     "+str$(ln),5)
220    prt(s+" /*  AR    DR    SR    RR    SL    OL    KS    ML  DT1  DT2  AME")
230    ln=ln+10
240    for j=1 to 4
250       s=right$("     "+str$(ln),5)+"     "
260       for i=0 to 10
270          s=s+right$("   "+str$(v(j,i)),3)+", "
280       next
290       if j=4 then prt(left$(s,len(s)-2)+"}") else prt(s)
300       ln=ln+10
310    next
320    prt(right$("     "+str$(ln),5)+" m_vset("+str$(k)+",v)")
330    ln=ln+10
340 next
350 if g then fwrites(chr$(&H1A),fn):fcloseall()
360 end
370 func str prt(s;str)
380    print s:if g then fwrites(s+chr$(13)+chr$(10),fn)
390 endfunc
400 func mkfl()
410    input "file name:",s
420    fn=fopen(s,"c")
430 endfunc
```

リスト2

リスト1 実行例

```
 100 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
 110 v={ 58,  15,   2,   0, 220,   0,   0,   0,   0,   3,   0,
 120 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 130     28,   4,   0,   5,   1,  37,   2,   1,   7,   0,   0,
 140     22,   9,   1,   2,   1,  47,   2,  12,   0,   0,   0,
 150     29,   4,   3,   6,   1,  37,   1,   3,   3,   0,   0,
 160     15,   7,   0,   5,  10,   0,   2,   1,   0,   0,   1}
 170 m_vset(1,v)
```

●きらきらした明るい方向に持っていきたい場合はトータルレベルを0に近づけていく。マルチプルの値を大きくしてみる。暗くしたい場合はその逆
●もっと，こもった感じの丸い音にしたい場合はトータルレベルの値を大きくしてみる
といった感じでしょうか。

ただしキャリアのトータルレベルは音量を決定するものなのでいじっても音色は変化しません。逆にマルチプルはキャリアだろうがモジュレータだろうが変更すれば音色が変化します。また，比率をそのままにして，このマルチプルをエディットすればオクターブを調整できます。

あと，音色データフォーマットの頭にAFというのがありますがこれは，

フィードバック×8+アルゴリズム

の値を書きます。このフィードバックも実は音質を構成するうえで重要な要素です。この値を大きくするとノイズのような「ジャン」といった感じになりますが，この効果が顕著に出てくるのはアルゴリズム4以上です。

音色がなんとなく似てきたら次にキャリアをエディットします。図3にエンベロープを構成するパラメータのイメージ的な関係を示しますが，この図を参考に目的とする音色の音量変化を頭に描き，これを大雑把でいいですから数値で打ち込んでみましょう。希望するものと違ったなら数値を微調整していきます。また，先ほどもいったとおりキャリアのトータルレベルは音量に相当しますので，たいていの場合は0（最大）でいいはずです。

ここまでで，かなり似た音色ができているはずです。さて，これから細かな音色調整，つまり先ほど「重要ではない」といいきった音質変化のエディットをしていきますが，とかく人間は数値を暗記するのが苦手です。万が一，エディットしていくうちに目的としている音色から掛け離れてしまったときに音色選びからやり直すのは馬鹿らしいので，ディスクにこの時点の音色をセーブしておきましょう。詳しくやると誌面がいくらあっても足りないのでここでもイメージ的なアドバイスをするにとどめます。

●音の硬さを変えたい

キャリア，モジュレータの区別なしに「アタックレイト（AR)」や「ディケイレイト（DRまたは1DR)」をエディットしましょう。また，鳴らすオクターブによっては「キースケーリング（KSまたはRS)」をエディットすることによっても音の硬さを変えられます。

●音の余韻を変えたい

これもキャリア，モジュレータの区別なしに「リリースレイト(RR)」をエディットしましょう。
と，こんな感じですが，上に挙げたのは一般的な場合にすぎません。

キャリアが複数あるアルゴリズムで各オペレータに役目を持たせてある音色については上のアドバイスどおりにエディットしても目的の音色にならない場合があります。たとえば，アルゴリズム4で作られたピアノの音色のなかにはオペレータ1，2で弦を叩いたときのノイズ，オペレータ3，4で実際の音階を鳴らしているものがあり，いくらノイズ発生部分の音質変化を修正しても音階となって耳で聞こえてくる音は変わりません。

さて，音色を似せる方法はこのへんで終わりにしますが，エディットしている途中で目的の音色とは違ったものができても，「これはいい音色だな！」と思ったらすぐさまセーブしましょう。後々その音が必要になってくるかもしれませんよ。

図2 アルゴリズム

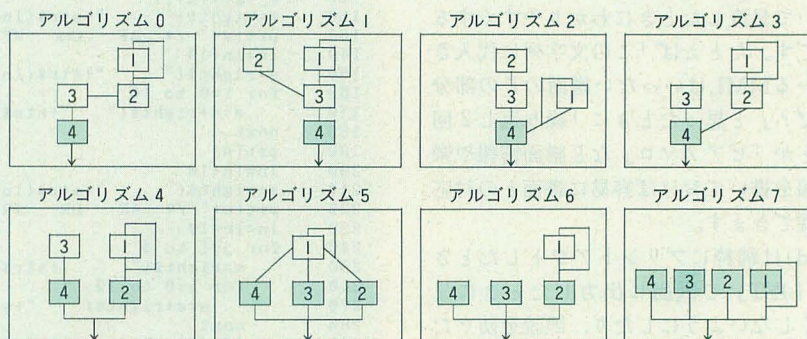
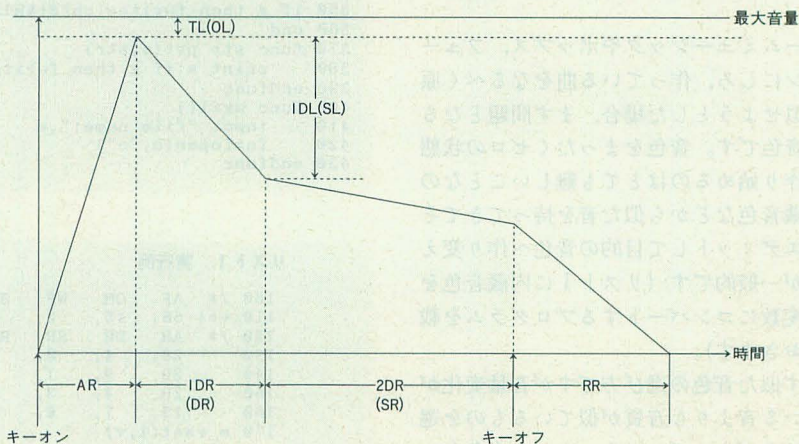



図3 エンベロープのパラメータ

読んで損はないMMLテクニック

ここで話すことはむしろテクニックというよりは，FM音源でミュージックプログラムを作る際の注意事項のようなものです。

音色切り替えのお話

さて，長い曲やチャンネル数が足りないときは音色切り替えが頻繁に行われるものです。この音色切り替えの際注意したいのが**切り替え直後に休符を置かない**ほうがよいという点です。切り替えた音色によっては次に続く休符のときに「ぷーん」などと音が鳴ってしまうことがあるからです。解決策としては音色切り替え直後に休符でなく音階を置く以外にありません（要するに休符のあとに音色切り替えをする)。

たとえば，

@1R4CDE

とせずに，

R4@1CDE

としたほうがよいということです。

また，音階を直後に置いた場合でも音色切り替えの終了した瞬間に「ブチ」という

ノイズが出る場合があります。こうなってしまった場合はそのチャンネルのリリースレイト全部とディケイレイト全部に15を書き込みましょう。チャンネル1の場合だったら音色切り替えの直前に,

Y224,255Y240,255Y232,255Y248,255

を書いておけば大丈夫のはずです。

レジスタナンバーや値の意味は今月号102ページに掲載されているFM音源レジスタマップを参照して理解してください。ちなみにレジスタナンバーの224,240,232,248は16進数で$E0_H$, $F0_H$, $E8_H$, $F8_H$,書き込んでいる値の255は16進数でFF_Hです。

「ぷーん」にしろ「ブチ」にしろ音色を切り替えを行う前の音色のリリースレイトが極端に長いことが原因のひとつです。特に「ブチ」のケースの対応策はかなり面倒臭いので音色のほうをエディットするのが賢明かもしれません。

また,新たに音源ドライバを作ろうとしている人はこの「ブチ」ノイズ防止のために音色切り替えの直前に必ずそのチャンネルの「1DL/RR」レジスタ全部にFF_Hを書き込むようにプログラムしましょう。

テンポが遅れ気味の場合

「OPMDRV.X」はかなり高速の割り込み周期で割り込みを行っているためテンポを早くしすぎたり,微小音長(@L1など)を多用するとテンポが遅くなることがしばしばあります(その点サン・ミュージカル・サービスのMUSICDRV.Xは立派)。これはいくつかのトラックのMMLの頭に,

@L1R (@L1でなくても奇数ならなんでもよい)

のようなダミーの休符(別に休符でなくてもいいが)を書いてやることである程度免れます。

理由は音源ドライバの作りをよく理解した人にでないと説明しにくいのでここで詳しいことは避けますが,まあ「割り込み周期を分散させ軽くする」ためとでも思ってください。この「@L1R」を書き込むトラックの選び方ですが,エコーパートなどに入れるといいでしょう。エコーパートなどは元々遅れて発音するパートですからね。

ひとつ注意しなくてはならないのはゲームミュージックのようなループする曲の場合,この「@L1R」は必ずループ演奏の開始指定コマンド「[DO]」などの前に置くということ。でないと演奏がループするたびにずれていってしまいますよ。

ハードLFOのお話

LFOディレイが設定できないとか,各チャンネルバラバラでLFOスピードを設定できない,波形も各チャンネルバラバラに設定できない,など,やたらに嫌われているハードLFOですがそんなに使えなくもありません。たとえば波形については普通の楽器のビブラート(ピッチモジュレーション)ならば三角波で十分ですし,スピードについては標準的な値に設定しておけばまず問題はないでしょう。ディレイですが,多少MMLが見にくくなるもののこれから話すテクニックを使えばなんとかなります。

まず曲中で使用するすべての音色について以下の処理を施してください(X68000のFM音源ではLFOのパラメータを音色ごとはおろか,チャンネルごとにさえ設定できないため)。ここでは音色データが配列変数Vに格納されているものとして話を進めます。

a) V(0,2)のSYC(シンクスイッチ)を0にする

b) V(0,4)のSPD(LFOスピード)は180~220の値にする

c) V(0,5)のPMD(ピッチモジュレーションデプス)は20~50の値にする

d) V(0,7)のPMS(ピッチモジュレーションセンシティビティ)を1以下にする

e) V(0,6)のAMD(アンプリチュードモジュレーションデプス),V(0,8)のAMS(アンプリチュードモジュレーションセンシティビティ)を適当な値(普通は0にしたほうがいい)にする

この処理が終わったらそれぞれの音色設定を行っている「m_vset(?,V)」命令の後ろに,

:V(0,7)=3以上7以下の適当な値を

付け加えます。これは音色パラメータのPMSを書き換えています。この書き換わった音色パラメータ配列変数を「m_vset」命令で適当な音色番号にセットしてください。これでやっと準備完了です。

パラメータを書き換える前にセットした音色をたとえば,いまここでは音色番号1とし,書き換え後のものを71とします。

@1C8&@71C8

と演奏してみましょう。どうです。音長の半分あたりから音が揺れ始めたと思います。つまり,@1で演奏した部分がLFOディレイに相当するわけです。文章だけではわかりにくいと思うのでリスト3に実行例を示しておきます。

しかし,この方法は先ほどもいったとおり非常に見にくいMMLとなること,また頻繁に音色切り替えを行わなくてはならないことが欠点といえば欠点です。

最後のわるあがき

本文で書き忘れたことや,行数をさいてまで説明することもない簡単なテクニック,注意事項をいくつか示して終わりにします。

●V15=@V125であるから最大音量は@V127である。

●バッキングなどの和音用の音色はアルゴリズム4以上で作成すると音が厚く聴こえる。

●和音を構成する1音1音,無理して違うパンポットに振り分ける必要はない(たとえば「CEG」の和音を鳴らす場合「C」を左,「E」を中央,「G」を右で鳴らしたりすると綺麗に聞こえないことがある)。

●Oh!Xのミュージックプログラムコーナーの選曲担当は私ではないのでゲームミュージックがいいとは限らない。

以上です。ご精読ありがとうございました。では皆さん,ミュージックプログラミングを楽しんでください。

リスト3 LFOディレイ

```
 10 dim char v(4,10)   :                      /*  Synth Lead
 20 /* AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
 30 v={48,  15,   2,   0, 210,  50,   0,   2,   0,   3,  0,
 40 /* AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 50    31,   4,   0,   2,   0,  33,   0,   2,   0,   0,   0,
 60    24,   0,   0,   2,   0,  24,   0,   2,   0,   0,   0,
 70    24,   0,   0,   2,   0,  18,   0,   8,   0,   0,   0,
 80    20,   0,   0,   6,   0,   0,   0,   2,   0,   0,   0}
 90 m_vset(1,v):v(0,7)=4:m_vset(71,v)
100 m_alloc(1,100):m_assign(1,1)
110 m_init():m_trk(1,"q8 o4 v15 y48,0 L4@1c&@71c@1d&@71d@1e&@71e@1f&@71f@1g&@7
1g"):m_play()
```

FM音源用MML記述テクニック

Yこそすべて

Orimo Naoki 織毛 直樹

ポルタメント，モジュレーション，そしてサンプリング音源制御……。あらゆる操作を可能にするという「Y」コマンドとはなにか？ FM音源を知り尽くした者だけに許される「Y」の秘密と超絶技巧の世界へ案内しよう。

X68000は楽器なのである

どうも，はじめまして。今回が初登場となります織毛です。これからもどーかよろしく。

さて，X68000も発売されてからすでに3年が過ぎ，ミュージック機能を扱う環境もだいぶ整ってきました。当初はふつうのユーザーには使用不可能だと思われていたサンプリング音源も，OPMAによって利用可能となり，さらにOPMDによって簡単にMIDIさえも扱えるようになりました。

これは，よーするに“豪華”な音が出せるようになったということで，ばーんばーんざいもの。しかし忘れるべからずは初心なり。基本は内蔵のFM音源だということは肝に銘じておかなければなりません。なぜなら，サンプリング音源はいくら生の音が出るとはいってもFM音源の引き立て役にすぎないし，MIDI音源のみによる演奏をするならば，わざわざ楽譜をMMLで打ち込む意味がないからです。つまり，全部の音源のバランスが取れてこそ，X68000という1台のマシンですべてを制御する意味が出てくるワケで，耳先（?）ばかりにとらわれてはいけません。

FM音源が使いこなせて初めて“サンプリング”や“MIDI”が生きてくるのですから，それらが互いに調和していなければ，単なる「あーとねーちゃーした井手らっきょ」にすぎないのです。

んがが，確かにFM音源の出せる音には限界があります。道行く人にはピコピコ音の延長だといわれてしまうかもしれません。どう逆立ちしようとも8つしか音が出ませんからね。でも，そんなところが逆に可愛いのであって，ちょっと打ち込んでみるというだけなら手軽な楽器としてのポテンシャルは十分に持っているのです（その実力はといえば，そんじょそこらのCASIOトーン顔負けなのだ）。

それにコンピュータのFM音源はほかのキーボード類とは違って，ユーザーと音源チップとのあいだが密着しているから，「思いどおりに操れる」という利点があります（実はこれが非常に大事なことなのだ）。むろん，それにはさまざまなノウハウが必要ですが，これはなかなかどうして「むふふ(パフパフ)」な関係なんですよね。

てなわけで，今回の音楽特集。

非常に高価な楽器となってしまったあなたのX68000の前でヘッドホンして，にたにたしながら，恐怖のYコマンドを並べようじゃあーりませんか。

音楽のススメ

あさてさて，実際にX68000ちゃんで音楽するワケですが，なにから始めるのがいいのでしょうか。まず，てっとりばやいのがOh!X LIVEコーナーに載っかっているプログラムをぽこぽこ入力すること。これだとプログラミングに悩むこともなく，根気さえあれば素晴らしい曲を愛機が奏でてくれます（**公理一**）。

一度も入力したことのない人はぜひ入力してみてください。最初はなんでも構いませんから。以前Oh!X LIVEコーナーに載った「代々木ゼミナール校歌」「この木なんの木」といったものが手頃なところですね。短いので10分もあれば十分でしょう。

絶対に笑えますよ。

でもって「そうか，ペケロクで音楽するのは面白いかもしれないな」と思ったのならしめたもの。今度はちょっと手の込んだOPMA，もしくはOPMD(MIDIなし版)用のわりと長めのものに挑戦してみてください。いわゆるサンプリングばしばし「進藤」，「立川」プログラムというヤツですね。たかがパソコンの音楽とはいっても，「うーむ，なかなかどうして」ものの，生音楽に引けをとらない曲が聞けるはずです。

この際，OPMDといった音源ドライバやサンプリングデータのように，なかなか手にすることが難しいものでも，6月号のおまけディスクにたぁーんと入っておりますので，ここぞとばかりに利用しましょう。ここまでの段階だったら本体とディスプレイとヘッドホンの基本システム（**公理二**）だけでこと足りますしね。ま，なによりも重要なのは実際に自分の手で入力したものを自分の耳で聞いてみるということです。

んで，「そーか，うちのペケロクはこんなにいい音がするのか」と，今度は自分でプログラムを作ってみようということになるワケですが，とっころがぎっちょんちょん。マニュアルの親切さとFM音源の簡潔さとがあいまって(そう，皮肉よ)，ここでほっぽりだしてしまう人が多いのが実態のようです。

西川の善さんが初心者向けの講座をやってくれているので，私は「だいたいのMMLはわかるんだけど，いま三歩いいものができないんだよなぁ」という，まだ初心者に片足を突っ込んでいる（おそらく大多数の）人のために“これさえ知っていればあなたも明日からミュージックプログラマー”という，とっておきのテクニックをご紹介することにします。

最近のOh!X LIVEの凝った作品にはすでに使われているものばかりですが「嵐のようなYコマンド」にうんざりして，解析するのをあきらめた人がほとんどではないかと思います。んでも，機械的にMMLを打ち込むのと，そのコマンドの意図を知ったうえで打ち込むのとではスピード的に雲泥の差があるわけで，知っていて損はしないハズ。うさんくさい世界ではありますが，FM音源の凄さを見直す機会になってくれればと思います。

では，まいりましょ。

まずは「ディチューン」

パソコンの曲を聞いて誰でも思うのが，「生の音にはかなわない」ということです。でも，FM音源がたった55個の数値で音の複雑な波形を決定しているのを考えれば，

●謎の裏チャンネル
X68000にはパンポット機能といって，音を真ん中と左右合わせて3方向に振る機能がついています。よーするにステレオ出力なワケです(公式七)。でも，完全的に音が別れてしまうので，うまく使わないと非常に聴きづらい，不自然な曲になってしまいます。

その出せる音のバリエーションは驚異的ともいえます。しかし，デジタル音源だからしかたないとはいえ，どうしても音のつぶつぶは粗くなる。ウスくなる。

そこで，「ウスいのならアツくしてしまえ」という単純な発想からきたのが“ディチューン”テクニック。

小学生の合唱コンクールがあると下手な者同士集められるのと一緒で，貧弱な音源でも同時発声をすれば豪華に聞こえるというものです。貴重なチャンネルを余計に使ってしまうのがこの技の欠点ですが，得られる効果はそれを補ってもあり余るほどで，けっこう頻繁に用いられます。

この際，同時に鳴らすチャンネル同士の出力周波数をわずかにずらしておくのがミソ。そのずれが心地よいうねりとなって聞こえてくることでしょう（**定理壱**）。

まず基本リストを実行したうえでリスト1を実行してみてください。リスト0と比べると違いは歴然でしょう。

なるへそ，と腕組みしたなら，次の技。

カラオケ技「エコー」

これはもう定型です。カラオケないし，風呂場で誰もが経験する偉大なる「エコー」効果のMML版といえるのがこの技。あまりにも有名でいまさら解説の必要はないかもしれませんね。方法はしごく簡単。エコーをかけたいフレーズを別のチャンネルで少し遅らせて音量を下げて追いかけさせれば，はい，おまちどお，いっちょーあがり。この際，定理壱を併用すると一層の効果が得られます。リスト2です。

さて，ここからが意外と知られていない，美味しい調味料にあたる部分。

先ほどの「少し遅らせて音量を下げて」というのは具体的にどれくらいなのでしょうか。もちろん，試行錯誤をして，聴感上もっとも適切で，その場面で求められている効果を探り出すのが望ましいのですが，一応基本パターンがあります。

1) 基本音長の1.5倍だけ遅らせて，
2) ボリュームの値を2だけ下げる

（定理弐）

これが基本です。ですから，まずこれに従って入力してみて，あとは聞きながら微調整というのがよいでしょう。あ，そうそう，ここでの基本音長というのは「その曲のなかでいちばん多く出てくる音長」のことです。つまり，8分音符が基本になっている曲だったら，付点8分音符分だけずらすということ。リスト2はこの基本どおりにMMLをセットしていますので確認してみてください。

あとは，細かくなるので具体例は示しませんが,「エコーパートの音色をメインよりも柔らかい音色にする」(反射音は反射する物体によってその波形が変わってくる。直接音と反射音が同一波形になるのは理想反射体の場合のみ)，「エコーパートを1音ごとに左右に振る」(限られた音場空間を最大限に活用する必要性より。別項にて詳説)，「メリハリのあるエコーパートの使用」(常に特定のパートにかけるのではなく，その

リスト0

```
 10 /*
 20 /*  <キホンリスト>   ニュースステーションテーマ
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     58,  15,  0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     19,   4,   3,   6,  11,  32,   1,   1,   0,   0,   0,
 90     20,   0,   0,   6,   0,  54,   1,   9,   0,   1,   0,
100     21,   0,   6,   6,   0,  58,   1,   1,   0,   0,   0,
110     22,   0,   4,   7,   0,   0,   1,   2,   0,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t120         o5       l32 @71 p3 q8  v14  y48,00 y15,0
190 b="frgrarbrrrarrrgr rrdrrrcrrrrrf1&f4
200 m_trk(1,a)
210 m_trk(1,b)
230 /*
240 m_play()
```

リスト1

```
 10 /*
 20 /*  <リスト1>   ディチューンヲカケタモノ
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     58,  15,  0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     19,   4,   3,   6,  11,  32,   1,   1,   0,   0,   0,
 90     20,   0,   0,   6,   0,  54,   1,   9,   0,   1,   0,
100     21,   0,   6,   6,   0,  58,   1,   1,   0,   0,   0,
110     22,   0,   4,   7,   0,   0,   1,   2,   0,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t120         o5       l32 @71 p3 q8  v14  y48,00 y15,0
190 b="frgrarbrrrarrrgr rrdrrrcrrrrrf1&f4
200 m_trk(1,a)
210 m_trk(1,b)
220 /*
230 /*
240 a="             o5       l32 @71 p3 q8  v14  y49,32 y15,0
250 b="frgrarbrrrarrrgr rrdrrrcrrrrrf1&f4
260 m_trk(2,a)
270 m_trk(2,b)
280 /*
290 m_play()
```

リスト2

```
 10 /*
 20 /*  <リスト2>   ディチューン+エコーヲカケタモノ   (チョットカケスギカナ・・・)
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     58,  15,  0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     19,   4,   3,   6,  11,  32,   1,   1,   0,   0,   0,
 90     20,   0,   0,   6,   0,  54,   1,   9,   0,   1,   0,
100     21,   0,   6,   6,   0,  58,   1,   1,   0,   0,   0,
110     22,   0,   4,   7,   0,   0,   1,   2,   0,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t120         o5       l32 @71 p3 q8  v14  y48,00 y15,0
190 b="frgrarbrrrarrrgr rrdrrrcrrrrrf1&f4
200 m_trk(1,a)
210 m_trk(1,b)
220 /*
230 /*
240 a="       r16.  o5       l32 @71 p3 q8  v12  y49,40 y15,0
250 b="frgrarbrrrarrrgr rrdrrrcrrrrrf1&f4
260 m_trk(2,a)
270 m_trk(2,b)
280 /*
290 m_play()
```

そこで「音場」について考えなければなりません。「音場」とは文字どおり，「音が鳴る場所，空間」のこと。X68000は3方向の出力しか持っていないので，この「音場」がきわめて狭いものになっているのはわかってもらえるはずですが，ここで「そういうもんなんだ」と開き直ってしまうのはちょっと早いもの。

「移動感」「裏チャンネルの効果的な利用」を追及すれば，狭いなりに豊かな音場を作り出すことが可能なので

時点でもっとも目立つパートにかけること）などの必須事項が挙げられます。

ほかにも，エコーに関しては「調子に乗って深くかけすぎない」(私も初めはこれにハマった)とか，「エコーに2チャンネル以上使わない」(なにせ少ないチャンネル。そんな無駄をするよりもっと有効な使い道がある）といった禁止事項もあります。

よーするに，こういったテクニックは音を引き立てるために使うのであって，それ自体が前に出てきてはならないのです。つまり，大切なのは，「単独で聞いたときの聞き心地」ではなくて「全体的な調和」を最優先することにあります。

「ふと気がついてみると，そこには美しいエコーのかかった旋律があった」，というのが達人のエコーといえるでしょう。

実はエコーにはほかに「たがいちがい技」（自分でつけるとなんてカッコ悪い名前になるんだろう）というのがあるのですが，楽譜の再構成という非常に神経を使う作業をしなければならなくて，一度MMLを打ち終えてからバグが見つかると，それ以後の部分をすべて書き直さなければならなくなるという「えげつない」シロモノなんですよね。

きわめて自然なエコー効果が得られるということと，エコーをかけているメロディが単音になったり和音になったりするときに（ピアノ曲によくある）真価を発揮するという点で優れモノではあるのですが，デバッグが異様に大変なので，ここでは準公式としてサンプルプログラムを挙げておくに留めます（でもこれだけはちゃんと入力してほしいな)。リスト3です。名前の由来は各チャンネルを別々に聞いてみればわかるようになってます。

センスのある人，自分でもっといい名前つけてね。

時代は「ポルタメント」なのであります

んじゃ，お次。「ポルタメント」というのは“音階と音階を滑らかにつないで演奏する”という，あれです。「ピッチベンド」と呼ばれることもあります。楽譜では音符同士が「ぶるぶる」でつながれているやつですね。エレキギターとかによく出てきます。また，ピアノでは絶対にできません（当たり前か)。

さて，これをMMLで記述するのですが残念ながらBASICのMMLには「最初の音符と最後の音符，そして長さを指定すると勝手にポルタメントにしてくれる」なんていう便利なコマンドはありません（善さんの記事を見てね)。ですから元からあるコマンド，定理を組み合わせてなんとかするしかしょうがない。

ここで，おもむろにBASICマニュアルをばさばさとめくってみます。すると，あったあった。

“&：前後の音符をつなげて演奏する”

これはなかなか使えそうですね。で，早速「O4C」から「O5C」まで1オクターブのポルタメントをこの“&”コマンドを用いて記述してみたのがリスト4です。

しかし，確かに「O4C」から「O5C」まで“&”で音はつながっているものの，単にドレミと音を並べただけなので，決して「なめらか」といえません。ま，人間の耳なんてあまり当てにならないもんで，テンポを速くすればこれでも滑らかに聞こえてしまいます（tの値を200にして実行してみてください)。でも，曲によってはながーくておそーいポルタメントが出てくることも，むろん，あるワケです。

リスト3

```
   10 /*  <リスト3>  「タカ゛イチカ゛イホウ」 ニヨル 「エコー」コウカ
   20 /*
   30 /*   オリシ゛ナルスコア 「イトシキリリモ」
   40 /*
   50 dim char v(4,10)={
   60 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN        きらきらてっきん
   70     60,  15,   3,   1, 237,   0,  88,   0,   1,   3,   0,
   80 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
   90     31,   0,   0,   0,   0,  43,   0,   6,   3,   0,   0,
  100     31,  12,   0,   3,   2,   0,   0,   2,   3,   0,   0,
  110     31,   0,   0,   0,   0,  60,   0,  15,   7,   3,   0,
  120     31,  12,   0,   3,   2,   0,   0,   8,   7,   0,   0}
  130 m_vset(71,v)
  140 m_init():for i=1 to 8:m_alloc(i,1000):next
  150 for i=1 to 8:m_assign(i,i):next
  160 str a[256],b[256],c[256],d[256],e[256],f[256]
  170 /*
  180 /*
  190 a="t100          o4       18  @71 p3 q3  v12  y48,00 y15,0
  200 b="errrrrrrer  r  rrer  rr  r r rr>br r   b r>r br rrr   r  rb rrr  r r rr
rrr rrr<<t92c  r  r  rr  r  t94r r  t92d t90r t89r  t88rt87r  t86rt85rt70r  t80g
r1
  210 m_trk(1,a)
  220 m_trk(1,b)
  230 /*
  240 /*
  250 a="              o4       18  @71 p3 q3  v12  y49,04
  260 b="crrrrrcrrr  r  rrrr  rr>>g-r rr rr r<  a-r r rr rrr  <c  rr rrr  r r rr
ccr rrr     r  r  r>>ar  r     r a     r    r    r    r  b    r   r   r    r
r1
  270 m_trk(2,a)
  280 m_trk(2,b)
  290 /*
  300 /*
  310 a="              o3       18  @71 p3 q3  v12  y50,08
  320 b="grrrrgrrrr  r  rrrr>>rb  r r rr rr r<<<e r r rr>brr   r  rr rra  r r rr
rrr rrr     f  r  r  rr  f      r r     g     r    r     r  r     r   r  b    r
r1
  330 m_trk(3,a)
  340 m_trk(3,b)
  350 /*
  360 /*
  370 a="              o3       18  @71 p3 q3  v12  y51,12
  380 b="rrrerrrrrr  r  rrrr  rg- r r rr rr r >>e r r rr rrr< <e  rr rrr  r r<dr
rrr rrr     r  r>>g  rr  r     r r<    b    r     r     r  r     r  g  r     r
r1
  390 m_trk(4,a)
  400 m_trk(4,b)
  410 /*
  420 /*
  430 a="              o3       18  @71 p3 q3  v12  y52,16
  440 b="rrdrrrrrrr  r  rrrr  ra  r r rr rr<d   r r r cr rrr   r  rr>crr  r r rr
rrr rrr     a  r  r  rr  r     r<c      r    r    r>>  a  r     r  r  r<<<  r
16gr2...
  450 m_trk(5,a)
  460 m_trk(5,b)
  470 /*
  480 /*
  490 a="              o2       18  @71 p3 q3  v12  y53,20
  500 b="rgrrrrrrrr  r<<rfrr  rr  r r>dr rr r   r r>a-rr rrr < g  rr rrr  r<c rr
rrr rrr     r>>c  r  rr  r     r r >    g    r     r     r  r<<   d   r   r     r
r1
  510 m_trk(6,a)
  520 m_trk(6,b)
  530 /*
  540 /*
  550 a="              o2       18  @71 p3 q3  v12  y54,24
  560 b="crrrrrrrrr<<g  rrrr  rr  r>d-rr rr r   r>e r rr rrr>  a  rr rrr<<b r rr
rrr rrr>>   f  r  r  rr  r<<   b r     r>   d    r     r  r     r  r  r<<   r
64br2..&r16..
  570 m_trk(7,a)
  580 m_trk(7,b)
  590 /*
  600 /*
  610 a="              o4       18  @71 p3 q3  v12  y55,28
  620 b="rrrrrrrdrr  r  rrrr  rd  r r rr rd-r   r r d rr rrr   r>>er rrr  r r rr
rrr<arr     r  r  r  rc  r     c r     r    r>   g    r   r     r  r  r<<<  r
32dr2..&r16.
  630 m_trk(8,a)
  640 m_trk(8,b)
  650 /*
  660 /*
  670 m_play()
```

す。「裏チャンネル」とは，本来なら3方向しかない出力の4番目に当たるチャンネルのことをいいます。2チャンネル使って左右から同じフレーズを（周波数を少しずらして）出力したときに，音が定位する位置を「裏チャンネル」と（私が勝手に）呼んでいるのです。実際に，ただパンポットを中央にあわせたものと同時に，なにか音を出して聴いてみてください。どちらも中央から音が出ているはずなのに，不思議とこの「裏チャンネル」と

そんなときは，音程が階段状に変化するせいで滑らかに聞こえないので，もっともっと微妙な音程を出して（あたかも）滑らか（なよう）に聞かせる必要がでてきます。そこで定理壱を見てください。な，ぬわんと，「音程を微妙にずらす公式」があるではありませんか！（わざとらしい）**（定理参）**

そんなこんなで，定理壱を利用した，より滑らかなポルタメントがリスト5。

いやー，Yコマンドがずーらずらで気色悪いですね。でも，滑らかさは聞いてのとおりで，手間をとるか完成度をとるかの違いになってきます。もっとも，どうせならYコマンド並べるのもやらせてしまえ，とプログラム中に関数を作ってしまうのもひとつの手ですが，RUNしてから音が出てくるまでにけっこう時間がかかるので私はあまり好きではありません。こんなのは好みの問題だけど。調子に乗ってばしばしといろんな機能つけすぎて「いやー，RUNしたの昨日なんだけどね」っていうのは，レイトレみたいでやだもん。

ぶるぶる「ビブラート」

このビブラートというのは「音声を細かくふるわせる奏法」（広辞苑第三版）とありますが，具体的には「音程を細かく上下させる」ことをいいます（ほかにも音量を細かく増減させるものがあるのですが，パソコンでは前者のほうが効果が顕著に表れるのでここでは扱いません）。

ところで，OPM（X1やX68000に搭載されているFM音源の名称）にはLFOという名の，音にビブラートをかける機能がついていて，これを使えば済みそうなものなのですが，「実質的に1チャンネルにしかかけることができない」，非常に有効的な機能，「発声してから一時おいてビブラートをかけ始める“ディレイ”がついていない」という欠点のために，「テンポが変わっても波形の周期が変わらない」，「音の減衰部分，つまりリリースにもビブラートがかかる」という利点があるにも関わらず，実際にはあまり使えない「宝の持ち腐れ機能」に成り下がってしまっています。つまり，ユーザー側でなんらかの工夫をしてやらなければならないのです**（定理四）**。

定義を振り返ってみますと，音程を「細かく」上下させるということなので，ここでも定理壱の公式が役に立ちます。よーするに，長い音符を細かくみじん切りにして，音程を少し上げては少し下げてを繰り返してやるのです。そうすれば図1のようなビブラートの「ぶるぶる」を再現できるでしょ？

これをプログラムしたのがリスト6。最後の音符にビブラートがかかっているのがわかると思います。この際，波形の1周期をひとまとめにして，リピートコマンドで繰り返すことによって短縮化を図っています。んが，どっちみちMMLが長くなってしまうのは避けられないので，曲の「締めどころ」だけでかけるとかして，なるべく少なく済ませる工夫をしてください。さもないと，素晴らしいドロ沼にずぶずぶと入っていくことになりますからね。初めのうちは「さわり」を出すだけでも十分でしょう。

真面目な話，いかに聞いている人を誤魔化すかというのも，実はきわめて重要なことなのです。

柔らかい便箋，「ソフトエンベロープ」

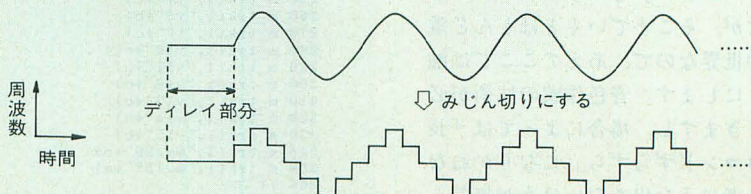
さて，前のビブラートやエコーもそうだったのですが，このあたりになってくるともはや楽譜に載っている情報ではなく，オリジナルだったら自分の感性に，コピーだったら元曲を聞き取る聴力に頼る部分が大きくなってきます。つまり，「ここはヤマ場だからエコーをたっぷりと効かせてビブラートも深ーく，うにょうにょさせせよう」とか，「うむむむ，ベース音とハイハットを音色切り替えのステレオ割り込みで1チャン

リスト4

```
 10 /*
 20 /*  <リスト4>  アライホ゜ルタメント     ネイロカ゛チカ゛ウノテ゛チュウイ
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN
 60     44,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     31,   0,   0,  15,   0,  21,   0,   1,   3,   0,   0,
 90     31,   0,   0,  15,   0,   0,   0,   1,   3,   0,   0,
100     31,   0,   0,  15,   0,  20,   0,   1,   7,   0,   0,
110     31,   0,   0,  15,   0,   0,   0,   2,   7,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t35          o4      164 @71 p3 q8  v14  y48,00 y15,0
190 b="c&c+&d&d+&e&f&f+&g&g+&a&a+&b&<c4
200 m_trk(1,a)
210 m_trk(1,b)
230 /*
240 m_play()
```

リスト5

```
 10 /*
 20 /*  <リスト5>  ナメラカナホ゜ルタメント
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     44,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     31,   0,   0,  15,   0,  21,   0,   1,   3,   0,   0,
 90     31,   0,   0,  15,   0,   0,   0,   1,   3,   0,   0,
100     31,   0,   0,  15,   0,  20,   0,   1,   7,   0,   0,
110     31,   0,   0,  15,   0,   0,   0,   1,   7,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t35          o4      @l1 @71 p3 q8  v14  y48,00 y15,0
190 b="y48,0c&y48,80c&y48,160c&y48,0c+&y48,80c+&y48,160c+&y48,0d&y48,80d&y48,1
60d&y48,0d+&y48,80d+&y48,160d+&y48,0e&y48,80e&y48,160e&y48,0f&y48,80f&y48,160f&y
48,0f+&y48,80f+&y48,160f+&
200 c="y48,0g&y48,80g&y48,160g&y48,0g+&y48,80g+&y48,160g+&y48,0a&y48,80a&y48,1
60a&y48,0a+&y48,80a+&y48,160a+&y48,0b&y48,80b&y48,160b&<y48,0c4
210 m_trk(1,a)
220 m_trk(1,b)
230 m_trk(1,c)
240 /*
250 m_play()
```

図1 ビブラートをYコマンドで表現する法

ネル分の「表チャンネル」を独立して，あたかも別のプレーンで鳴っているかのように聞こえます。これも余計なチャンネルを喰う，というデメリットがあるのですが，なかなか使えそうな技でしょ？　たとえば，表チャンネルをメインパートに，「裏チャンネル」をエコーパートに割り当てて「エコー」（公式二）をかけてみると，「うーんと広ーい空間」が頭の中に出現します。さらに定理六も適用させると，

ネルに収めているな，流石はコナミの曲だ」といった感じです。

初めのうちは見落としがちだった音の細かい表情が見えてくるのも，こういった楽譜には書かれていない情報を自分の手で入力しだすあたりからです。

「ソフトエンベロープ」というのもそういった音の微妙な表情のうちのひとつで，ファンファーレとかにありがちな「出た音がいったん減衰して，またじわじわと立ち上がってくる」(むむむ，音を文章で表現するのは難しいな）というような時間的音量変化がある音を再現するときに使うテクニックです。ブラス系の音によくある効果ですね。

もっとも，ゲームミュージックなどではこういった表情を細やかに出しているものは少なく，生音楽（なにげなく使ってるけど変な日本語）を元にプログラミングするとき，特に注意して聞いて（読んで）ください。

ま，テクニックとはいってもその発想さえ手にしてしまえばたいしたもんではありません。前のビブラートのところで使った「音のみじん切り」を利用して音符のあいだにボリュームコマンドをバシバシ挟んでいけばいいのです。とにかく聞いてみてください。リスト7です（**定理五**)。

音が小さくなってから大きくなるのがわかると思いますが，ここで大事なのは，いかにも人間が楽器を吹いているかのように聞かせるということ。

また，この段階まできてしまうと完全に人それぞれの感性の問題で，公式なんかないから試行錯誤を繰り返して“勘”を養うしかありません。Oh!X LIVEコーナー常連の人のようにミュージックプログラミングに長けている人とそうでない人の差が出てくるのもこのあたり。

ある程度のものができるようになったらテクニックを盗むつもりで他人のプログラムを研究していくのも上達の近道かもしれません。実は，私もそうしてきたひとりだったりすのだよん。ふふふ。

ちなみに，時間的音量変化だけでなく，時間的音色変化というのを再現する「ソフトモジュレーション」というテクニックもあるのですが，そこまでいくとほとんど重箱つつきの世界なので，あえてここでは触れないことにします。音色作成の技術が必要になってきますし，場合によっては「長い長いYコマンドずらずら」になりかねないので初心者は手を出さないほうが無難といえるでしょう。

1チャンネルで二度おいしい「リバーブ」

さあ，いよいよ小賢しい領域に入ってきました。1チャンネルのみでエコーに近い効果が得られる「リバーブ」技です。さっきからしつこく，チャンネルを無駄なく使うようにいってきてるので，まさにうってつけの技といえます。

この技は「FM音源を休ませてはいけない」という発想からきました。つまり，休んでる暇があったらどっかで音を出してろ，というワケ。うーむ，残酷。

どんな曲のフレーズでも途中に休符というものがありますよね（たまに全然ない変な曲もあるけど)。で，当然のことながら，FM音源はその休符で休みをとります。そこに目をつけてなにかに利用できないかと考えるワケです。

でも，なにせ休符のやってくるタイミングは不規則。なんか別の楽器音を出すにしても，その音を出すタイミングが限定されていては曲になりません。特に元曲がある場合には，そのフレーズをアレンジしてし

リスト6

```
 10 /*
 20 /*  <リスト6>  ビブラートヲカケタモノ
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     19,   4,   3,   6,  11,  32,   1,   1,   0,   0,   0,
 90     20,   0,   0,   6,   0,  54,   1,   9,   0,   1,   0,
100     21,   0,   6,   6,   0,  58,   1,   1,   0,   0,   0,
110     22,   0,   4,   7,   0,   0,   1,   2,   0,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,1000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 /*
170 /*
180 a="t120         o5        132 @71 p3 q8  v14  y48,00 y15,0
190 b="frgrarbrrrarrrgr rrdrrrcrrrrrf4&@l1
200 c="|:12y48,40fy48,80fy48,120fy48,160fy48,120fy48,80fy48,40fy48,0fy48,220ey
48,180ey48,140ey48,100ey48,140ey48,180ey48,220ey48,0f:|y8,0
210 m_trk(1,a)
220 m_trk(1,b)
230 m_trk(1,c)
240 /*
250 /*
260 a="        r16.   o5        132 @71 p3 q8  v12  y49,40
270 b="frgrarbrrrarrrgr rrdrrrcrrrrrf4&@l1
280 c="|:12y49,80fy49,120fy49,160fy49,200fy49,160fy49,120fy49,80fy49,40fy49,0f
y49,220ey49,180ey49,140ey49,180ey49,220ey49,0fy49,40f:|y8,1
290 m_trk(2,a)
300 m_trk(2,b)
310 m_trk(2,c)
320 /*
330 m_play()
```

リスト7

```
 10 /*
 20 /*  <リスト7>  サラニ「ソフトエンベロープ」ヲカケタモノ  SRガカワッテイルノデチュウイ
 30 /*
 40 dim char v(4,10)={
 50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
 60     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
 70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
 80     19,   4,   2,   6,  11,  32,   1,   1,   0,   0,   0,
 90     20,   0,   0,   6,   0,  54,   1,   9,   0,   1,   0,
100     21,   0,   3,   6,   0,  58,   1,   1,   0,   0,   0,
110     22,   0,   0,   7,   0,   0,   1,   2,   0,   0,   0}
120 m_vset(71,v)
130 m_init():for i=1 to 8:m_alloc(i,2000):next
140 for i=1 to 8:m_assign(i,i):next
150 str a[256],b[256],c[256],d[256],e[256],f[256]
160 str g[256],h[256],j[256],k[256],l[256],m[256]
170 /*
180 /*
190 a="t120         o5        132 @71 p3 q8  v14  y48,00 y15,0
200 b="frgrarbrrrarrrgr rrdrrrcrrrrrl32v14f&v13f&v12f&v11f&v10f&v9f&v8f&v7f&@l
1
210 c="y48,40fy48,80fy48,120fy48,160fy48,120fy48,80fy48,40fy48,0fy48,220ey48,1
80ey48,140ey48,100ey48,140ey48,180ey48,220ey48,0f
220 m_trk(1,a)
230 m_trk(1,b)
240 m_trk(1,"v6"+c)
250 m_trk(1,"v7"+c)
260 m_trk(1,"v8"+c)
270 m_trk(1,"v9"+c)
280 m_trk(1,"v10"+c)
290 m_trk(1,"v11"+c)
300 m_trk(1,"v12"+c)
310 m_trk(1,"v13"+c)
320 m_trk(1,"v14"+c)
330 m_trk(1,"v15"+c)
340 m_trk(1,"@v126"+c)
350 m_trk(1,"@v127"+c)
360 /*
370 m_play()
```

「うーみーはー，ひろいーなー，おーきーなー…」。
本当に信じられないくらいの広がりが出てくるので，ぜひ試してみてください。
さて，問題がひとつ。「ヘッドホンをつけないと裏チャンネルが認識できない」ということ。これは外部スピーカーだと「左の音は左耳だけで，右の音は右耳だけで」聴くことができないからです（モノラルの内蔵スピーカ

まうわけにもいきませんからね。

逆にいうと，オリジナル曲であれば休符のタイミングをうまく調節して，そこに別のフレーズを（リズム楽器などがやりやすい）さくさく埋め込めば，あら不思議，1チャンネルで2つのフレーズが鳴ってしまうというものも作れます。

ただ，これには作曲や編曲の技術，音場空間を効果的に利用する技術，さらに，いかにも2チャンネルで演奏しているかのように聞かせる「誤魔化し」の技術が必要になってくるので，かなり高度な技であることは否めません。興味のある人には，コナミ曲のリズムパート，ファミコン曲の一部（マッドシティとか）のフレーズなどが参考になるでしょう。

さて，その休符部分を別のフレーズに使うのが難しいとなると，あとはなんらかの“効果”に回すしかありません。

有名なのが「繰り返し音によるエコー効果」。やまびこのように音(単音だけでなく短いフレーズでも可）を小さくしながら何度も繰り返すことによってエコーを得るという方法です。

しかし，確かにこれだと1チャンネルでエコーがかかるのですが，まとまった長さの休符がないとそれらしく聞こえない，何度も音を繰り返すために多用すると旋律が濁ってくる，という理由でいま0.5歩(いま一歩まではいかない）見送りです。一応サンプルは作っておきましたので聞いてみてください。リスト8です。

んでもって，ここで登場するのが「リバーブ」技（なんとまあ，手前ミソ）。図2を見てください。上が休符になにも小細工をしないときの音量変化で，下が休符のときに音量を半分にしたものです。どうでしょう。FM音源が休符の部分で余韻として立派に働いているのがわかるでしょうか。

このときに注意しなければならないのが「リバーブをかける音色のキャリアのリリースレートを長めにとる」必要があるということです。

今回は基本的に，音色作成の知識がなくてもわかる技ばかりにしたのですが，これだけはしかたがありません。キャリアというのがなんだかわからない人や，リリースレートってなんじゃらほい(死語)？　という人は「リストの網掛けになっている部分を機械的に3，ないし4に書き換える」ことだと考えてください。一応，それでなんとかなります。

具体的なMMLの記述は実際にリスト9を見て覚えてください（**定理六**)。

ステレオ化に関する諸注意

さて，いよいよ最後。締めくくりの意味でいままでとは少し趣向を変えて，ミュージックプログラムを組む際に，上手い人とそうでない人の差がいちばん出てくる「音源のステレオ化」について簡単に説明することにしましょう。

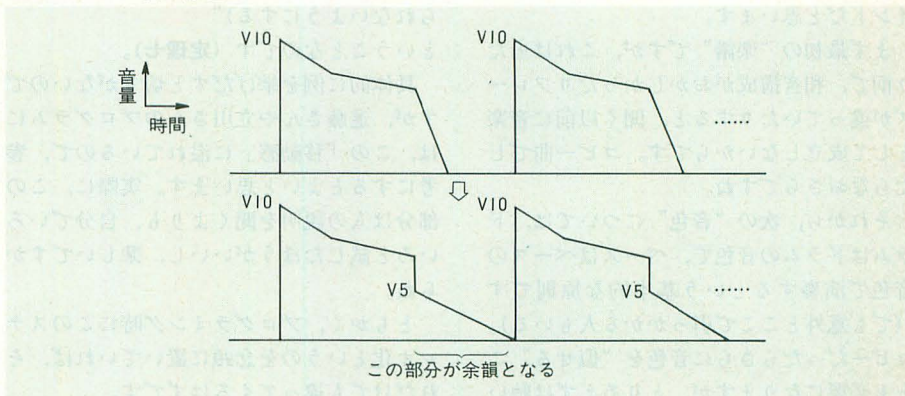



図2 リバーブ技の原理

リスト8

```
  10 /*
  20 /*  <ﾘｽﾄ8>  ﾔﾏﾋﾞｺｴｺｰ   ﾈｲﾛｶﾞ ｶﾜｯﾃｲﾙﾉﾃﾞ ﾁｭｳｲ
  30 /*
  40 dim char v(4,10)={
  50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
  60     60,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
  70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
  80     31,   0,   0,   0,   0,  30,   0,   8,   3,   0,   0,
  90     18,  12,   9,  12,   5,   0,   0,   8,   3,   0,   0,
 100     31,   0,   0,   0,   0,  20,   0,   4,   7,   0,   0,
 110     31,  12,   9,  12,   5,   0,   0,   8,   7,   0,   0}
 120 m_vset(71,v)
 130 m_init():for i=1 to 8:m_alloc(i,1000):next
 140 for i=1 to 8:m_assign(i,i):next
 150 str a[256],b[256],c[256],d[256],e[256],f[256]
 160 /*
 170 /*
 180 a="t120       o1      l16 @71 p3 q8  v14  y48,00 y15,0
 190 b="v14av13av12av14av13a<v14egafv13fv12fv14dv13dv12d>v14av13av14gv13gv12gv1
4gv13gv14b-<df>a-v13a-v12a-v14a-v13a-<v14ea-v13a-v12a-v11a-v10a-v9a-v8a-v7a-v6a-
v5a-v4a-v3a-v2a-v1a-v0a-r8.
 200 m_trk(1,a)
 210 m_trk(1,b)
 220 /*
 230 m_play()
```

リスト9

```
  10 /*
  20 /*  <ﾘｽﾄ9>  ﾘﾊﾞｰﾌﾞ        RRﾉｱﾀｲｶﾞ ｶﾜｯﾃｲﾙﾉﾃﾞ ﾁｭｳｲ
  30 /*
  40 dim char v(4,10)={
  50 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN  Sax
  60     58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
  70 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
  80     19,   4,   3,   3,  11,  32,   1,   1,   0,   0,   0,
  90     20,   0,   0,   3,   0,  54,   1,   9,   0,   1,   0,
 100     21,   0,   6,   3,   0,  58,   1,   1,   0,   0,   0,
 110     22,   0,   4,   3,   0,   0,   1,   2,   0,   0,   0}
 120 m_vset(71,v)
 130 m_init():for i=1 to 8:m_alloc(i,1000):next
 140 for i=1 to 8:m_assign(i,i):next
 150 str a[256],b[256],c[256],d[256],e[256],f[256]
 160 /*
 170 /*
 180 a="t120       o5      l32 @71 p3 q8  v14  y48,00 y15,0
 190 b="fv8rv14gv8rv14av8rv14bv8rrrv14av8rrrv14gv8r rrv14dv8rrrv14cv8rrrrrv14f4
&@l1
 200 c="|:12y48,40fy48,80fy48,120fy48,160fy48,120fy48,80fy48,40fy48,0fy48,220ey
48,180ey48,140ey48,100ey48,140ey48,180ey48,220ey48,0f:|y8,0@v0r
 210 m_trk(1,a)
 220 m_trk(1,b)
 230 m_trk(1,c)
 240 /*
 250 m_play()
```

まずひとつあるのが，Oh!X LIVEコーナーに投稿されてくるプログラムのほとんどが「モノラル」のまんまであるという悲しい事実。

もちろん，常連さんのものは工夫の限りを凝らしてあって，素晴らしい“広がり”があるものばかりなのですが，どうやらまだ大多数の人が「ステレオ化が音源の貧弱さを補う必殺技だ」ということを知らない

ーは論外)。つまり，ヘッドホンだと両側から出力した音は“真ん中”に聞こえるのに対して，外部スピーカーだと両側から出した音が“両側”から聞こえてしまうので，結果的に音が散漫としてしまうのです。「音質がいい」以外に，こんな理由で私はヘッドホンを推しているのですが，まだちゃんとしたヘッドホンでX68000の曲を聴いている人は少ないのが実情。みんな，ヘッドホンを買うんだ。　(織)

ようです。というよりも，ちゃんとヘッドホンで曲を聞いている人が少ないのかもしれません。

ここで，私の独断と偏見によりますが，ミュージックプログラムの「勝負どころ」を挙げておきますと，

1．楽譜
2．音色
3．音場

と，この3つが押さえなくてはならないポイントだと思います。

まず最初の“楽譜”ですが，これは当たり前で，和音構成がおかしかったりフレーズが違っていたりすると，聞く以前に音楽として成立しないからです。コピー曲でしたらなおさらですね。

それから，次の“音色”については，ドラムはドラムの音色で，ベースはベースの音色で演奏するという基本的な原則です（でも意外とここで引っかかる人もいる）。コピーだったらさらに音色を“似せる”ことも必要になりますが，とりあえずは聴いていて不自然じゃなければ十分。

そして最後の“音場”がまさに「ステレオ化」に当たる部分で，「コンピュータ曲丸出し」にしないようにするために，いちばん気を遣う部分なのです。

ここで早とちりしてはいけないのが，ステレオ化というのが「単にチャンネルを左右に振り分けることではない」ということ。重要なのは，

“チャンネルに最大限の移動感を持たせて少ない音源を多く見せかける（音源をステレオにあちこち振って聞く人が追いかけられないようにする）”

ということなのです（**定理七**）。

具体的に例を挙げだすと切りがないのですが，進藤さんや立川さんのプログラムには，この「移動感」に溢れているので，参考にするとよいと思います。実際に，この部分は人の説明を聞くよりも，自分でいろいろと試したほうがいいし，楽しいですからね。

ともかく，プログラミング時にこのステレオ化というのを念頭に置いていれば，それだけでも違ってくるはずです。

あーでもない，こーでもない，といいながら「Pコマンド」をがりがり書き込んでくださいね。

＊　　＊　　＊

えっと，駆け足ながら「MMLさえ知っていればすぐ使えるテクニック」を紹介してきました。が，まだまだ音色に関するテクニック，サンプリングを活用するときのテクニックなど，ミュージックプログラムを作るうえで大事なものがいろいろとあります。

「え，サンプリングにテクニックなんてあるんかいな？」と思う人もいそうですが，冒頭でも説明したように，サンプリングの「よい音」に惑わされて，ただガンガンと鳴らせばいいってもんではありません。みてくれだけの“かつら”としてではなく，FM音源にプラスαをする“増毛法”としてサンプリングを使うには，いろいろなノウハウが必要なのです。

まあ，そういったものはまたの機会にしますので，とにかくいまはいろんな曲をプログラムして腕を磨いてください。もちろん，よいものが出来たらOh!X LIVEまで投稿するのを忘れないよーに。では，みんながんばって。

●「公理定理公式」

公理一

「プログラムは間違えずに打つ」

当然といえば当然だが，ミュージックプログラムには「エラーにならないバグ」がたくさん出てくるから注意が必要である。

「心を込めて打ち込む」

こらこら，馬鹿にするんじゃない。こーすると打ち間違えも減るってものなのだ。半分嘘みたいな話だけど，半分は冗談なんだよん。

公理二

「音楽やるならヘッドホンを使う」

ハミダシで説明しているとおり，ウラチャンネルとかを微妙に聞き分けるにはヘッドホンは必需品。それも高品質のものをひとつ持っておきたいところだ。音楽用のボードを買ったと思えば安いってモノ。

定理壱

「周波数を微妙にずらすにはYコマンドを使え」

これがYコマンドの基本。あとはOPMAやOPMDでサンプリングを鳴らしたり，音をキーオフするときくらいにしか使わない。ま，知ってる人も多いでしょ。

公式：「Y 47+n,4m」
　n：チャンネル番号
　{n | 1≦n≦8}
　m:ずらし加減。1で最小。63でほぼ半音上がる
　{m | 0≦m≦63}

定理弐

「エコーは別チャンネルで同フレーズを追いかけさせろ」

これも有名な技。一応公式は次のとおり。

公式：エコーパートの設定
　「Vn-2 rm.」
　n：音量の値
　{n | nはメインパートの音量}
　m：音長の値
　{m | mは曲の基本音長}

定理参

「ポルタメントは音をなるべく細かく区切って間にYコマンドを挟め」

これは定理一の応用編。Yコマンドを使わなくても済むならそのほうがいいに決まっていて，下手に細かくしすぎないのもテクのうち。ゼンジー西川さんのZMUSIC.FNCを打ち込むか，Yコマンド並べるか。あ，どっちも大変なのね。

公式：一般型を書いてもごちゃごちゃするだけだから省略だあね。

定理四

「ビブラートなどは繰り返し記号で短縮化を図れ」

Yコマンドでいちばん手に負えないのがこれ。ずらずらと並んでいても意味がさっぱりわからないということもある。入力するときは要注意。

公式：これだけは経験的処理が必要なので公式はなし。あえていうならば「人のプログラムを見て真似する」こと。

定理五

「ソフトエンベロープは下手に@Vコマンドを使うな」

@Vコマンドの音量変化は滑らかだからいいけど，対数変化だから，聴感上Vコマンドのほうが扱いやすい。

公式：「Vn」
　n：音量の値。数が増えるほど大きくなるが，0でも無音ではない。
　{n | 0≦n≦15}

定理六

「リバーブの余韻部分だけをステレオに振ると効果が高い」

実は「リバーブ」と「エコー」を同時にかけるとより美味しい。いわゆる定理の組み合わせというヤツ。音色のリリースを長めにとることを忘れずに。

公式：「Vl m Vl÷2 r Vl」
　l：そのチャンネルのボリュームの値
　m：演奏すべき音符
　r：演奏すべき休符
　{l | 0≦l≦15}

定理七

「ステレオ化ではとにかく『移動感』で勝負せよ」

パートによる割り振りも大切だが，差が出てくるのは移動感があるかないかというところ。リズム系の楽器は特に顕著にこの差が出る。ステレオに振るだけで音が躍動的になるからね。

公式：「Pn」
　n：音の左右真ん中の出力の指定
　1：左　2：右　3：中央
　{n | 1≦n≦3}

ここに挙げた公式はあくまでも目安にすぎません。時と場合と気分によってその都度補正をしてくださいな。

（ただし出てくる変数はすべて整数）

多彩な表現をマクロにする外部関数

ZMUSIC.FNC

Nishikawa Zenji 西川 善司

一般的なMMLで，ポルタメントやモジュレーションといった表現を使おうと思ったら，それこそ常人の理解を超えた大量のYコマンドをちりばめなければなりません。もっと簡単に豊かな表現を，というわけで生まれたのがZMUSIC.FNCです。

ボンジュール，西川善司デース。いやぁ，もう秋ですね。外を見れば，ほんのり黄色く色づいた銀杏の葉がヒラヒラと舞い落ちているではありませんか。道行く人の服装も白系統から茶系統へ。おや，いま，庭を横切ったのは夏毛から冬毛に生えかわった野ウサギ！　そうかぁ，もう冬への支度を始めた動物もいるんだなぁ……。

……なんちて。実はこの原稿を書いているのは，夏も真っ盛り，8月中旬。アツイアツイ。言えばいうほど暑くなる，この暑さを袋に詰め込んで，冬に開放する……，そんなエアコンできませんかね，電気代もかからないし。それはさておき，秋です(まだいってる)。秋といえば芸術の秋。芸術といえば音楽です。今月は「電子音楽術入門」ということで私は「こいつぁ，便利，必殺MMLマクロコマンド，ZMUSIC.FNC」をお届けします。

恐怖のYコマンド

リスト1，2を見てください。これは，Oh!X LIVEの常連「進藤君」と「立川君」のリストの一部です。凄いですね。なにが凄いって，「Yコマンド」の嵐がですよ。説明すると，リスト1はOPMのレジスタ「キーフラクション（KEY FRACTION)」に周期的な値を書き込むことによってビブラート効果を作り出しています。リスト2はキーフラクションへ1方向的な値を書き込んで行うテクニック，ポルタメント（ピッチベンド）です。

このキーフラクションというレジスタは，音程を微妙にずらすためのもので，0～63の範囲の値を書き込むことによって最大半音上までずらすことが可能です（実際にレジスタへ書き込むデータは4倍しなくてはならない)。このレジスタを用いたテクニックとしては，前出のモジュレーション，ポルタメントのほかに，ディチューンと呼ばれるものがあります。これは単に，互いに音程の微妙にずれた音を2声重ねて発声するだけなのですが，実に厚みのある豊かな音になります。

こうした，キーフラクションを使ったテクニックは，もはやFM音源を使う人間には当たり前になってきているのですが，ひとつ問題があります。それは，プログラムリストとして掲載されたときに，入力しようとする人が思わず月に向かって吠えてしまうということです。

「わをーん！　Y48,0C&Y48,40C&Y48,80C&Y48,120C&……，ああーんもう勘弁してくだしゃーい」

Oh!Xのミュージックプログラムを入力していて，こう叫んでしまった人も少なくないはずです。これらのYコマンドの嵐がなにをやっているのか理解している「その筋人間」ならまだしも，ごくたまにミュージックプログラムを入力するような一般人(?)にとっては拷問以外のなにものでもないことでしょう。

実は，先ほど挙げた例にしても公差40の等差数列（なつかしー）というれっきとした規則性があるわけですし，なんとかコンピュータ側にこれらと同様なことをさせることができないか，誰しも思うはずです。そう，これが今回作成したZMUSIC.FNCのテーマなのです。

ZMUSIC.FNCの動作

さて，ポルタメントにしろなんにしろ，新たなMMLコマンドを増やすにはFM音

リスト1　モジュレーション（進藤君の場合）

```
   1 /*進藤君の「メタルホーク」より…
16000 p(0)="[d.c.]r1r1    @71 v12  l12 o3 p3 q8 y53,16   [coda]
16010 p(1)="v11c&v9fv12b-<e-6&@14|:4y53,16e-&y53,220d&y53,16e-&y
53,68e-&:|y53,16y8,5112dc>b-@13
16020 pol(16,-34,16,6,10):p(2)=l+"&
16030 p(3)="@14|:6y53,16c&y53,144>b&y53,16<c&y53,128c&:|y53,1611
2>g-b-g-<d-6&@14|:4y53,16d-&y53,200c&y53,16d-&y53,88d-&:|y53,16y
8,5>
16040 pol(16,-24,12,6, 5):p(4)=l+"&y53,16|:3y53,16g-&y53,220f&y5
3,16g-&y53,68g-&:|y53,16y8,5112
16050 p(5)="<fe-d-c>b-a-<c>b-a-<@14|:3y53,16a-&y53,240g&y53,16a-
&y53,48a-&:|y8,5112>
```

リスト2　ポルタメント（立川君の場合）

```
   1 /* 立川君の「サンダークロス」より
12020 a="       [d.c.] [coda] o3 @12  @1 p3 q8   v14 y49,20
12030 bb="y49,020c& y49,050c& y49,080c& y49,110c& y49,140c& y49,
170c& y49,200c& y49,230c&
12040 cc="y49,020c+&y49,050c+&y49,080c+&y49,110c+&y49,140c+&y49,
170c+&y49,200c+&y49,230c+&
12050 dd="y49,020d& y49,050d& y49,080d& y49,110d& y49,140d& y49,
170d& y49,200d& y49,230d&
12060 ee="y49,020d+&y49,050d+&y49,080d+&y49,110d+&y49,140d+&y49,
170d+&y49,200d+&y49,230d+&
12070 ff="y49,020e& y49,050e& y49,080e& y49,110e& y49,140e& y49,
170e& y49,200e& y49,230e&
12080 gg="y49,020f& y49,050f& y49,080f& y49,110f& y49,140f& y49,
170f& y49,200f& y49,230f&
12090 hh="y49,020f+&y49,050f+&y49,080f+&y49,110f+&y49,140f+&y49,
170f+&y49,200f+&y49,230f+&
12100 jj="y49,020g& y49,050g& y49,080g& y49,110g& y49,140g& y49,
170g& y49,200g& y49,230g&
12110 kk="y49,020g+&y49,050g+&y49,080g+&y49,110g+&y49,140g+&y49,
170g+&y49,200g+&y49,230g+&
12120 ll="y49,020a& y49,050a& y49,080a& y49,110a& y49,140a& y49,
170a& y49,200a& y49,230a&
12130 mm="y49,020a+&y49,050a+&y49,080a+&y49,110a+&y49,140a+&y49,
170a+&y49,200a+&y49,230a+&
12140 nn="y49,020b& y49,050b& y49,080b& y49,110b& y49,140b& y49,
170b& y49,200b& y49,230b&<
```

源ドライバの改造または作り直しが必要になってきます。しかし，改造というのはかなり危険な行為で，プログラムも非常に組みにくくなるため今回は遠慮しました。そこで，今回着眼したのはX-BASICの外部関数MUSIC.FNCです。

このMUSIC.FNCという外部関数は音源ドライバではなく，BASICプログラムから受け取ったパラメータを単にOPMDRV.X（本体同梱のFM音源デバイスドライバ）へ送っているだけです。MMLをセットするM_TRKという命令にしてもMMLの書いてある文字変数をBASICから受け取りそのままOPMDRV.Xへ転送しているだけなのです。これをそのままでなく，加工してから送るのがZMUSIC.FNCです。

つまり，送るべき文字列中にあらかじめ決めておいた「新たなMMLコマンド」が存在したならば，それを従来のMMLコマンドに変換してから送ってやろうというわけです。

具体的な例を挙げて説明しましょう。ド（C）からミ（E）へのポルタメントを，

(C，E)

と記述すると決めたとします（新たなMMLコマンド）。ZMUSIC.FNCはその「新たなMMLコマンド」を確認すると「Yコマンドの嵐（従来のMMLコマンド）」へと変換し，それからOPMDRV.Xへ転送するのです。何十文字も打ち込まなくては表現できなかったポルタメントもいまの例でいくと，たった5文字ですんでしまうのです。

まあ，早い話，M_TRK命令の機能拡張といったところですかね。しかし，面倒臭いことはコンピュータにさせちまえ，という基本思想にのっとった味のあるアイデアに私は，しばしお尻にできたおできの痛みも忘れ酔いしれたものです。

入力方法

リスト5のダンプリストをマシン語入力ツール（6月号付録のmac.x）から入力するか，またはリスト6のソースをエディタから入力しアセンブルしてください。アセンブルからリンクまでの手順を初心者のために詳しく説明します。まず，

A>ED ZMUSIC.S

でエディタを起動します。リスト6を入力終了後，[ESC]＋[E]でエディタを抜けて，

A>AS ZMUSIC.S

とアセンブラでアセンブルしてください。また，本リストはIOCSCALL.MACとDOSCALL.MACをインクルードしますので，これらのファイルをカレントディレクトリ（この場合はソースリストなどが存在するディレクトリ）へコピーしておくか，

A>AS /I IOCSCALL.MACなどの存在するディレクトリ ZMUSIC.S

のようにしてアセンブルしてください。アセンブル時，何かしらのエラーが出た場合は，

A>ED ZMUSIC.S

として再びエディタを起動して，打ち込んだリスト6を修正してください。

無事アセンブルを終了すると「NO FATAL ERROR (S)」というメッセージが出ます。このメッセージを確認したら，

A>LK ZMUSIC.O

としてリンクしてください。

これで，ZMUSIC.Xというファイルができたはずです。

注）アセンブラ，リンカ，IOCSCALL.MAC，DOSCALL.MACは，「C compiler PRO-68K」や「THE 福袋 V2.0」に入っています。また，IOCSCALL.MAC，DOSCALL.MACは本誌6月号の付録ディスクにも収録されています。

外部関数の組み込み

さて，アセンブル，リンクを終了するとZMUSIC.Xというファイルができますがこれを，

A>REN ZMUSIC.X ZMUSIC.FNC

としてZMUSIC.FNCにリネームしてください。そして，このZMUSIC.FNCを，

A>COPY ZMUSIC.FNC ¥BASIC2

のようにして，自分のシステムディスクのBASICのディレクトリへコピーしてください。

注）初代X68000のシステムディスクではBASIC2ではなくBASICというディレクトリになっています。各自システムディスクを確認してください。

次にBASICのコンフィギュレーションファイルを変更します。

A>CD BASIC2

でBASICのディレクトリへ入り，

A>ED BASIC.CNF

でコンフィギュレーションファイルを読み込んでください。

```
FREE  =128
WIDTH =64
BEEP  =ON
CAPS  =OFF
FUNC  =GRAPH
  ⋮
```

のようなものが画面に出てくるはずです。一番下の行に，

FUNC =ZMUSIC

を付け足し[ESC]＋[E]でエディタを終了してください。これで準備完了です。

A>BASIC

でZMUSIC.FNCを組み込んだBASICが起動します。

ZMUSIC.FNCのコマンド詳細

ZMUSIC.FNCを組み込むと，M_TRK2という命令が使用可能になります。従来のM_TRKも使用可能ですので従来の音楽プログラムも同じシステムで実行可能です。それでは，M_TRK2で追加されたコマンドを解説していきたいと思います。

Kn（ディチューン）

このコマンドの書かれたチャンネルの音程を微妙にずらします。nは0≦n≦63の範囲の値をとり，n=63で半音上となります。通常は0～15を与えてやるといいでしょう。ちなみにこのコマンドは，

コンフィギュレーションファイルを選択してBASICを起動

新しい外部関数が発表となるたびにディスクを用意してそれを組み込むようなBASICシステムディスクを作るのはどうもスマートでありませんね。知っている人も多いかと思われますが，任意のBASICコンフィギュレーションファイルを組み込んでBASICを立ち上げる方法があるのでこれを紹介しておきます。

たとえば，ZMUSIC.FNCを組み込むように設定したコンフィギュレーションファイルがBASIC_ZM.CNFという名前でAドライブのBASIC2というディレクトリにセーブしてあるとします。そこでこのファイルに書かれた環境でBASICを立ち上げるには，

A>BASIC /cA:¥BASIC2¥BASIC_ZM.CNF

とします。「/c」というスイッチの後ろにコンフィギュレーションファイルの名前を書いて起動すればいいのです。「/c」を設定しないとBASIC.CNFというコンフィギュレーションファイルを勝手に選択して起動します。私は，

BASIC /cA:¥BASIC2¥BASIC_ZM.CNF

をZM.BATというバッチファイルにしてZMUSIC.FNCを組み込んだBASICが，

A>ZM

で立ち上がるようにしています。

Y48＋チャンネル番号(0～7),n×4がOPMDRV.Xへ送られます。BASIC起動時，デフォルトとして0がセットされています。また，値を省略すると以前に設定した値を使用します。

例)

K10 K

¯ _ （相対ボリューム）

ボリュームを相対的に変化させることができます。「¯」はボリュームアップ，「_」はボリュームダウンです。「¯_」の後ろには0～127の値を書くことができ，省略すると以前に与えられた値が使用されます（デフォルトは3です)。たとえば，

例)

@V120¯3C¯D¯E_F

というMMLをM_TRK2を用いてセットしたとすると以下のように変換されます。

@V120@V123C@V126D@V127E @V124F

ここで，「C」は120＋3＝123で@V123で演奏されるのは明らかですね。「C」の後ろに「¯」がありますが値がありません。この場合は，先ほど述べたようにその直前に使用された値3が起用され123＋3＝126となります。また，「D」の後ろにまた「¯」がありますので126＋3＝129，としたいところですが，FM音源では@V127が最大値ですので「E」は@V127で演奏されます。これと同様に相対ボリュームコマンドを用いて値が最小値である0を超えてマイナスになったときは@V0に変換されます。ちなみに「¯」はシフトキーを押しながら「^」，「_」はシフトキーを押しながら「かな」の「ろ」が印刷してあるキーを押すと出てきます。

($k_1$oct k_2)lng （ポルタメント）
oct＝“＜＞，”

音程をk_1からk_2へ絶対音長lngの長さで変化させます（絶対音長とは「@L」で全音符を192とした場合の音長設定のことです。詳しくはBASICマニュアルのM_TRKの項をどうぞ)。octはオクターブスイッチのことですが次の例を見ていただくと意味がわかると思います。

例)

(C<E－)48

は「C」から**ひとつ上のオクターブ**の「E－」まで，絶対音長48，つまり192/48＝4，4分音符の長さで滑らかに変化させます。「＜」を「＞」に変えると「C」からひとつ下のオクターブの「E－」までのピッチダウン，また，「，」にすると同じオクターブの「E－」までのピッチアップとなります。lngは省略すると以前の「L」または「@L」コマンドで設定したデフォルトの音長が使用されます。octは省略不可で，必ず「＜＞，」のいずれかを書いてください。またこのコマンドの後ろに「&」を書くことによって「タイ」「スラー」が表現可能です。

例)

(C,E) 24&E4.

＝n （モジュレーションスイッチ）

後述するモジュレーション効果を有効にするか無効にするかのスイッチです。「＝0」でオフ「＝1」でオンです。「＝」のあとの値を省略すると以前に与えたスイッチの逆を実行します。「＝1」のあとに「＝」を実行したとすればオフとなるわけです。

このコマンドはZMUSIC.FNCの内部ワークへスイッチを設定するだけでOPMDRV.X側へはなにも出力しませんが「＝1」の状態から「＝0」と設定したときだけ，（標準に戻すという意味で）以前に「Kコマンド」で設定された「キーフラクション」を書き込みます。

例)

＝1

^n_1,spd,dly （モジュレーション）

モジュレーションのパラメータを設定します。n_1は振幅で0≦n_1≦126が設定可能です。63が半音に相当するので最大上下2半音分音の揺らぎを表現できます。spdはモジュレーションのスピードで1≦spd≦16の範囲で設定可能です。16がもっとも速く，13前後が実用的な値です。

最後のパラメータdlyはディレイタイムで，どのくらいの時間をおいてからモジュレーションをかけ始めるかを設定できます。1≦dly≦8の値が設定範囲となっており，1と設定すると音の鳴り始めから，4とすると音長の真ん中くらいからモジュレーションがかかり始めます（図2）。

パラメータは省略不可能ですが，デフォルトでは^15,14,4が設定されています。

なお，このコマンドは「＝」コマンド同様，ZMUSIC.FNCの内部ワークにパラメータを格納するだけで，OPMDRV.Xへはなにも出力しません。

例)

^10,13,2

@Qn

Qコマンドというのが初めから用意されているのは多くの方がご存じでしょう。これは，1音中で実際に音を出している時間を1/8～8/8の割合で設定するものでした。@QはこのQを細かく絶対音長の値で設定する命令なのですが，後ろから設定するというのが多少変わっています。

いま，@Q10を設定したとして，4分音符を鳴らしたとするとしましょう。4分音符は絶対音長でいくと48です。発音時間は48－（@Qで設定した値10)＝38となります。つまり，@Qで設定した音長分だけ発音を早く終えるというわけです。

ただし，もし，@Qで設定した値より音長が短い場合は@Qは無効になります。たとえば@Q30としたとき，8分音符（絶対音長＝24）を鳴らしたとすると24－30＝－6となって発音時間がなくなってしまうのではなく，音長分24まるまる発音するということです。

図1 @VとVの関係

V	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
@V	0	87	90	93	95	98	101	103	106	109	111	114	117	119	122	125

図2 DELAYの説明

デフォルトでは0が設定されており@Qは無効となっています。また，@Qを使用する場合は「Q」コマンドのほうは「Q8」と設定しておいてください。

例)

　　@Q10

ZMUSIC.FNC使用上の注意

ZMUSIC.FNCを組み込むと，M_TRK2という命令が使用可能になるわけですが，標準のM_TRK命令同様にトラックバッファの確保と，チャンネルのアサインをしてからでないと正常に動作しません。説明するまでもないかと思いますがトラックバッファの確保はM_ALLOC，チャンネルアサインはM_ASSIGNで行います（詳細はBASICマニュアルを参照してください)。

さて，M_TRK2はチャンネルのアサインに関してひとつ制約があります。それは**FM音源のチャンネル番号ch(1～8)は必ずトラック番号ch+8×n(n=0，1，2，……)へアサインされていなくてはなりません。**つまり，FM音源のチャンネル1～8はトラック1～8，9～16，17～24，……へアサインされなくてはならないということです。なんか，字で書くとたいそうな制約に見えますが，たいしたことはありません。多くの皆さんが普通に使っている，

　　for　i=1 to 8：m_alloc (i,5000)

　　：m_assign (i,i)：nest

でいいわけですから。

初めに断ったようにZMUSIC.FNCは新しく設けたコマンドを従来のMMLに展開する「アセンブラ」のマクロ的な動作をします。命令詳細のところでいくつかの命令についてはどんなMMLに展開されるのかを示しておきましたが，ここでは「ポルタメント」と「モジュレーション」について詳しく説明したいと思います。

●ポルタメント

ポルタメントはリスト3に示すBASICプログラムのようなアルゴリズムのもとでMMLを生成します。基本的に「@L1」つまり絶対音長「1」で，目的の音程までキーフラクションを駆使して滑らかに音程を変化させるMMLをはき出すのです。

リスト3中央に「経験的処理」というコメントがありますが，これはたとえば，「C」から「D」へのポルタメントのように，2つの音程がさほど離れていないときなど，絶対音長「1」でMMLをはき出すには及ばない（つまり，そんなに細かい精度を必要としない）場合，絶対音長2以上を用いて掃き出すようにプログラムを制御する部分です。

具体的には2つの音程が半音以内の場合は「@L4」，2半音以内の場合は「@L3」，3半音以内の場合は「@L2」でMMLを生成します。生成したMMLの後ろに必ず「デフォルト」の音長を出力するのでユーザーはいちいち音長を再設定する必要はありません。

　　(C<E)

のように2つの音階の間にオクターブコマンドを挿入することによって1オクターブ以上のポルタメントも可能になるのは前にお話ししたとおりですが，このコマンドの実行後デフォルトのオクターブ値も変化します。たとえば，

　　O4 (C<E)

を実行するとオクターブは「5」になるわけです。

リスト3は，

　　(C>A) 48

を実行した場合とほぼ同じMMLを画面に出力します。

●モジュレーション

リスト4がZMUSIC.FNCのモジュレーションアルゴリズムをBASICで表したものです。モジュレーションは，必ず絶対音長「@L1」で展開されます。リスト4は，

　　^32,10,1 =1 B4

のときのMMLとほぼ同じものを画面に表示します。

リスト4では@L1でズラーッと音符を書いていますがZMUSIC.FNCではモジュレーションの周期を自動的に計算し「|：」「：|」の繰り返しコマンドを用いてMMLの短縮化を行います。

次にディレイのパラメータはどうMMLに反映されるのかを説明します。面倒臭い話は嫌いだという人は図2さえ理解しておいてくれれば，これから話すことは読み飛

リスト3　ポルタメントテスト

```
  10 /*ポルタメントテストプログラム
  20 float l=48
  30 float kf=0         /*detune
  40 str q="c",cm=">",r="a"
  50 str k(12)={"c","c+","d","d+","e","f","f+","g","g+","a","a+
","b","<c"},ti
  60 float a,n=0,y,s,b,ll,sg,bs,be
  70 int lw,ch=0        /*ch=channel
  80 w1=(instr(1,"c c+d d+e f f+g g+a a+b",q)+1)/2
  90 w2=(instr(1,"c c+d d+e f f+g g+a a+b",r)+1)/2
 100 if cm=">" then w2=w2-12
 110 if cm="<" then w2=w2+12
 120 bs=kf:be=abs(w2-w1)*63+kf
 130 sg=sgn(w2-w1)
 140 b=abs(be-bs)
 150 ll=1:lw=1:ti=""
 160 if l>15 then {     /*経験的処理
 165 ti="&"
 170 if (l mod 2)=0 and b<192 then ll=2:lw=2
 180 if (l mod 3)=0 and b<128 then ll=3:lw=3
 190 if (l mod 4)=0 and b<64  then ll=4:lw=4
 200 }
 210 w1=w1-1
 220 if sg<0 then w1=w1-1:y=63+kf else y=bs
 230 if y>63 then y=y-63:w1=w1+1
 240 if w1<0 then w1=11:print">";
 250 if w1>11 then w1=0
 260 print"@L"+str$(ll);" ";
 270 if l-1=0 then print"error":end
 280 s=int(b/(l-1))
 290 a=int(256*(b mod (l-1))/(l-1))
 310 for i=1 to l
 320 lw=lw-1:if lw=0 then lw=ll:prt()
 330 n=n+a
 340 if n>255 then y=y+sg:n=n-255
 350 y=y+s*sg
 360 next
 365 if ti="&" then print chr$(&H1D)+" "
 370 end
 380 func prt()
 390 if y>63 then y=y-63:w1=w1+1
 400 if y<0 then y=y+63:w1=w1-1
 410 if w1<0 then w1=11:print">";
 420 if w1>11 then w1=0:print"<";
 430 if y<>63 then print"y";str$(48+ch);",";str$(y*4);k(w1);ti;
else print"y";str$(48+ch);",0";k(w1+1);ti;
 440 endfunc
```

図3　リスト3の実行例

```
>@L2 y48,236b&y48,204b&y48,172b&y48,140b&y48,108b&y48,76b&y48,44
b&y48,12b&y48,232a+&y48,200a+&y48,168a+&y48,136a+&y48,104a+&y48,
72a+&y48,40a+&y48,8a+&y48,228a&y48,196a&y48,164a&y48,132a&y48,10
0a&y48,68a&y48,36a&y48,4a
```

ばしていただいて結構です。

音長Lをまず8で割りそれをディレイのパラメータ値から1引いたものと掛け合わせ，これをディレイの音長とします。この音長を「@L」コマンドで書き，その後ろに与えられたキーコードを書きます。あとは，リスト4のような「@L1」のモジュレーションMML部分がくる，という寸法です。少しわかりにくいと思うので具体例を用いてもう一度説明します。いま，

^15,14,4 =1 C4

をM_TRK2を使ってセットしたとしましょう。「C4」がモジュレーションとして展開されることになるわけですが，この音符の音長は4分音符，つまり絶対音長48です。この48を8で割ります。48/8＝6。モジュレーションパラメータのディレイとして4が与えられていますが，これから1を引きます。4－1＝3。この3と先ほどの6を掛けた3×6＝18がディレイ音長となります。つまり，

@L18C&|:n …モジュレーション展開部… :|

となります。モジュレーションとして展開された部分は絶対音長に換算して30となっているはずです。これで18＋30＝48，4分音符分無事演奏されるわけです。

モジュレーションをオンにするとMMLを使用するうえでひとつ制約が出てきます。それは「|:」「:|」**を使用した繰り返しコマンドが使用できなくなる**ということです。これはすでにモジュレーションの展開部分で「|:」「:|」を用いてしまっているためで，多重ループのできない繰り返しコマンドを持つOPMDRV.Xを使う以上しかたのないことです。つまり，モジュレーション・オンの状態で，

|: C:|

とすると，これを展開すると，

|: |:…モジュレーション展開部…:| :|

となり外側の「|:」「:|」は無視されてしまうのです。

ならば，繰り返し部分も展開してしまえ，という人もいることでしょう。つまり，

|: CDE :|

なら，

CDE CDE

と，してしまえということですね。それは私も考えたのですが，OPMDRV.Xに対して一度文字列を送ってしまうともう，それっきりで，外部関数側からは操作することができないのです。このため複数の文字列に渡った「繰り返し」が再現不可能となってしまいます。

また，このモジュレーションは極端にバッファを食うため上のように展開したとすると，まず，アっという間にトラックバッファが一杯になり，まず実用になりません。これらの理由からモジュレーション・オンのときは「|:」「:|」を使用できません。ご了承ください。

トラックバッファについて

さて，繰り返し言ってきたとおり，この2つのマクロコマンド，「ポルタメント」と「モジュレーション」はいずれも微小音長のMMLに変換するため，多用するとかなりトラックバッファを消費します。多用するときは普段より多くトラックバッファを確保しましょう。また，いくらトラックバッファを多く取りたいといってもOPMDRV.Xの登録時にそのマキシマムはすでに決定されていますから，どうしても，もっと！というときはCONFIG.SYSを書き換え，システムをもう一度立ち上げ直すしかありません。初心者のために具体的に示すとしましょう。

A>ED CONFIG.SYS

として，エディタにCONFIG.SYSを読み込んでください。画面に，

FILES＝15
BUFFERS＝20
BELL＝BEEP.SYS
DEVICE＝PRNDRV.SYS
⋮

と出てくるはずです。画面から「DEVICE＝¥SYS¥OPMDRV.X」という1行を探し出してください。見つけたらOPMDRV.Xの後ろに「#」を書いてさらにその後ろに希望するトラックバッファの容量を記述してください。180Kバイトなら，

DEVICE＝¥SYS¥OPMDRV.X #180

とします。変更したら[ESC]＋[E]でエディタを終了してください。あとはリセットスイッチを押してシステムを立ち上げ直せばOKです。

リスト4 モジュレーションテスト

```
 10 /*モジュレーションテストプログラム
 20 str o(15)={"b-","b","c","c+","d","d+","e","f",
 30             "f+","g","g+","a","a+","b","c","c"}
 40 str k="b"
 50 float a,b,sw,sg=1
 60 int n1=32 /*振幅
 70 int spd=10    /*speed
 80 int lg=48     /*length
 90 int oct=0,ch=1    /*channel
100 w1=(instr(1,"c c+d d+e f f+g g+a a+b",k)+1)/2+1
110 kf=0      /*detune
120 spd=17-spd:if spd=0 then print"error"
130 a=n1/spd
140 b=kf
150 m1=n1+kf  /*上限
160 m2=-n1+kf /*下限
170 print"@L1 ";
180 for i=1 to lg
190 prt()
200 b=b+a*sg
210 if b>m1 or b<m2 then sg=-sg:b=b+a*sg*2
220 next
230 if oct=-1 then print"<" else if oct=1 then print">"
240 end
250 func prt()
260 n=0:bb=b
270 if bb>0 then {
280 while bb>63
290 bb=bb-63:n=n+1
300 endwhile
310 } else {
320 while bb<0
330 bb=bb+63:n=n-1
340 endwhile   }
350 print "y";str$(48+ch);",";str$(bb*4);
360 if w1+n<2 and oct<>-1 then print">";:oct=-1
370 if w1+n>13 and oct<>1 then print"<";:oct=1
380 if w1+n>1 and oct=-1 then print "<";:oct=0
390 if w1+n<14 and oct=1 then print ">";:oct=0
400 print o(w1+n);
410 endfunc
```

図4 リスト4の実行例

```
@L1 y49,0by49,16by49,32by49,48by49,64by49,80by49,96by49,112by49,
128by49,112by49,96by49,80by49,64by49,48by49,32by49,16by49,0by49,
236a+y49,220a+y49,204a+y49,188a+y49,172a+y49,156a+y49,140a+y49,1
24a+y49,140a+y49,156a+y49,172a+y49,188a+y49,204a+y49,220a+y49,23
6a+y49,0by49,16by49,32by49,48by49,64by49,80by49,96by49,112by49,1
28by49,112by49,96by49,80by49,64by49,48by49,32by49,16b
```

M_TRKとM_TRK2

M_TRKとM_TRK2は共存が可能です。ですから，たとえば，あるチャンネルはマクロコマンドは使わないからM_TRKのみを使い，また別のチャンネルはM_TRK2でがりがりMMLをセットするといったことが可能なのです。しかし，同一チャンネルをM_TRK, M_TRK2をとっかえひっかえシーケンスすることは避けてください。

オマケ

ZMUSIC.FNCを組み込むと，M_TRK2のほかにM_SWITCHという命令も使用できるようになります。これは「おまけ」に付けたコマンドで，M_TRK2命令にセットしたMMLをファイルに書き出すためのスイッチ命令です。

M_SWITCH(1)

とすると，以後，M_TRK2にセットされた文字列や，マクロコマンドによって展開された文字列がカレントドライブのカレントディレクトリにZMUSIC.TMPというファイル名で書き出されます。当然のことながらディスクやドライブの準備ができていないとエラーになります。出力されるファイルは「OPMファイル」フォーマットに準拠しており，エディタやCOMMAND.XのTYPEコマンドで見ることができます。このスイッチを用いた完全な「OPMファイル」の作り方は別項を参照してください。

サンプルプログラム

サンプルプログラムとして，セガ・メガドライブのゲーム「スーパー忍」のメインテーマ「THE SHINOBI」をお届けします。作曲は「イース」「ソーサリアン」「スキーム」「ボスコニアン」……と次々に名曲を生みだすスーパーコンポーザー古代祐三氏によるものです。

ZMUSIC.FNCを組み込んだBASICを立ち上げてからリスト7を入力，RUNで実行してください。「トラックバッファが足りません」といったエラーが発生した場合には前述したCONFIG.SYSを書き換える方法でトラックバッファを大きくしたOPMDRV.Xを再起動してから実行してください。

＊

ZMUSIC.FNCはミュージックプログラムを入力する人の立場，制作する人の立場に立って作られたものなのですが，大きな問題があります。それは，このZMUSIC.FNCを入力しなくてはその恩恵にあずかることができない点です（当たり前だな……)。また，今後，このZMUSIC.FNCを使用したミュージックプログラムがOh!X LIVEのコーナーに投稿され，発表されたとしたら，当然のことながらZMUSIC.FNCを持っていない人，今月号を持っていない人は，そのプログラムを実行できません。これはミュージックデータに関するOh!Xの最大のテーマ「ミュージックデータの共有性」から大きくそれるものとなります。この辺については読者の皆さんのご意見を聞きたいところです……。

図5 FM音源レジスタマップ（参考）

レジスタマップ(00_H〜$1F_H$)

レジスタ番号	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
01_H	TEST						LFO RESET	
08_H		OPマスク				CH No.		
		OP4	OP3	OP2	OP1			
$0F_H$	NE			NFREQ				
10_H	CLK A_1							
11_H							CLK A_2	
12_H	CLK B							
14_H	CMS		FRESET		IRQ EN		LOAD	
			B	A	B	A	B	A
18_H	LFRQ							
19_H	F	PMD or AMD						
$1B_H$	CT_2	CT_1					W	

(20_H〜FF_H)

	CH0	CH1	CH2	CH3	CH4	CH5	CH6	CH7	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
	20_H	21_H	22_H	23_H	24_H	25_H	26_H	27_H	R	L	FEED BACK			ALG		
	28_H	29_H	$2A_H$	$2B_H$	$2C_H$	$2D_H$	$2E_H$	$2F_H$		KEY CODE						
	30_H	31_H	32_H	33_H	34_H	35_H	36_H	37_H	KEY FRACTION							
	38_H	39_H	$3A_H$	$3B_H$	$3C_H$	$3D_H$	$3E_H$	$3F_H$		PMS					AMS	
OP1	40_H	41_H	42_H	43_H	44_H	45_H	46_H	47_H		DT1			MUL			
OP2	50_H	51_H	52_H	53_H	54_H	55_H	56_H	57_H								
OP3	48_H	49_H	$4A_H$	$4B_H$	$4C_H$	$4D_H$	$4E_H$	$4F_H$								
OP4	58_H	59_H	$5A_H$	$5B_H$	$5C_H$	$5D_H$	$5E_H$	$5F_H$								
OP1	60_H	61_H	62_H	63_H	64_H	65_H	66_H	67_H		TL						
OP2	70_H	71_H	72_H	73_H	74_H	75_H	76_H	77_H								
OP3	68_H	69_H	$6A_H$	$6B_H$	$6C_H$	$6D_H$	$6E_H$	$6F_H$								
OP4	78_H	79_H	$7A_H$	$7B_H$	$7C_H$	$7D_H$	$7E_H$	$7F_H$								
OP1	80_H	81_H	82_H	83_H	84_H	85_H	86_H	87_H	KS			AR				
OP2	90_H	91_H	92_H	93_H	94_H	95_H	96_H	97_H								
OP3	88_H	89_H	$8A_H$	$8B_H$	$8C_H$	$8D_H$	$8E_H$	$8F_H$								
OP4	98_H	99_H	$9A_H$	$9B_H$	$9C_H$	$9D_H$	$9E_H$	$9F_H$								
OP1	$A0_H$	$A1_H$	$A2_H$	$A3_H$	$A4_H$	$A5_H$	$A6_H$	$A7_H$	AMS-EN			IDR				
OP2	$B0_H$	$B1_H$	$B2_H$	$B3_H$	$B4_H$	$B5_H$	$B6_H$	$B7_H$								
OP3	$A8_H$	$A9_H$	AA_H	AB_H	AC_H	AD_H	AE_H	AF_H								
OP4	$B8_H$	$B9_H$	BA_H	BB_H	BC_H	BD_H	BE_H	BF_H								
OP1	$C0_H$	$C1_H$	$C2_H$	$C3_H$	$C4_H$	$C5_H$	$C6_H$	$C7_H$	DT2			2DR				
OP2	$D0_H$	$D1_H$	$D2_H$	$D3_H$	$D4_H$	$D5_H$	$D6_H$	$D7_H$								
OP3	$C8_H$	$C9_H$	CA_H	CB_H	CC_H	CD_H	CE_H	CF_H								
OP4	$D8_H$	$D9_H$	DA_H	DB_H	DC_H	DD_H	DE_H	DF_H								
OP1	$E0_H$	$E1_H$	$E2_H$	$E3_H$	$E4_H$	$E5_H$	$E6_H$	$E7_H$	IDL				RR			
OP2	$F0_H$	$E1_H$	$F2_H$	$F3_H$	$F4_H$	$F5_H$	$F6_H$	$F7_H$								
OP3	$E8_H$	$E9_H$	EA_H	EB_H	EC_H	ED_H	EE_H	EF_H								
OP4	$F8_H$	$F9_H$	FA_H	FB_H	FC_H	FD_H	FE_H	FF_H								

リスト5　ZMUSIC.FNCダンプリスト

(6424バイトでセーブ)

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 00 00 00 00 17 84  : 9B
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 01 54 00 00 00 00  : 55
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  00 00 00 6C 00 00 00 A6  : 12
0048  00 00 00 E2 00 00 00 E2  : C4
0050  00 00 00 E2 00 00 00 A6  : 88
0058  00 00 00 E2 00 00 00 E2  : C4
0060  00 00 00 40 00 00 00 52  : 92
0068  00 00 00 64 00 00 00 00  : 64
0070  00 00 00 00 00 00 00 00  : 00
0078  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  48 55 01 0A 00 00 17 E6  1C93

0080  6D 5F 74 72 6B 32 00 6D  : BC
0088  5F 73 77 69 74 63 68 00  : F1
0090  00 00 00 00 00 5A 00 00  : 5A
0098  00 60 00 04 00 08 80 01  : ED
00A0  00 04 80 01 00 00 00 E4  : 69
00A8  00 00 0B 8E 42 39 00 00  : 14
00B0  0D 7C 22 3C 00 00 01 F0  : D8
00B8  93 C9 70 80 4E 4F 22 40  : 4B
00C0  22 3C 00 00 01 F0 70 80  : 3F
00C8  4E 4F B3 FC 00 FE 00 00  : 4A
00D0  65 0C 23 FC FF FF FF FF  : 8C
00D8  00 00 0E 34 4E 75 42 B9  : 00
00E0  00 00 0E 34 4E 75 41 F9  : 3F
00E8  00 00 12 EC 47 F9 00 00  : 3E
00F0  13 2C 49 F9 00 00 13 6C  : 00
00F8  70 07 20 FC 00 30 6A 00  : 2D
-----------------------------------
SUM:  C4 45 75 6B 52 7F 7A 1F  5FB9

0100  20 FC 03 00 01 00 26 FC  : 42
0108  00 0F 00 03 26 FC 00 03  : 37
0110  00 00 38 FC FF FF 51 C8  : 4B
0118  FF E2 42 39 00 00 0D 7D  : E6
0120  4E 75 4E 75 23 CF 00 00  : 78
0128  0E 14 61 00 0B 78 22 6F  : 97
0130  00 16 24 2F 00 0C 23 C2  : 5A
0138  00 00 0E 0C 61 2A 24 39  : 02
0140  00 00 0E 0C 4A 39 00 00  : 9D
0148  0D 7C 67 06 61 00 0A 94  : F5
0150  60 0C 72 06 70 F0 4E 4F  : E1
0158  4A 80 66 00 0B 56 42 80  : 53
0160  61 00 0B 38 42 80 4E 75  : 29
0168  45 F9 00 00 13 84 47 F9  : 15
0170  00 00 12 EC 49 F9 00 00  : 40
0178  13 6C 4B F9 00 00 13 2C  : 02
-----------------------------------
SUM:  EB F9 13 1D 79 F4 2F AB  73BA

0180  7E 00 1E 02 53 07 02 07  : 01
0188  00 07 13 C7 00 00 0D 68  : 56
0190  20 07 E3 08 D9 C0 10 07  : C2
0198  E7 48 DB C0 E7 0F D7 C7  : 5E
01A0  12 19 10 01 02 00 00 DF  : 1D
01A8  0C 00 00 41 65 08 0C 00  : C6
01B0  00 47 63 00 02 30 0C 00  : E8
01B8  00 4B 67 00 09 60 0C 00  : 27
01C0  00 4C 67 00 08 FE 0C 00  : C5
01C8  00 56 67 00 09 1C 7C 01  : 5F
01D0  0C 01 00 40 66 2E 0C 11  : FE
01D8  00 4C 67 00 09 28 0C 11  : 01
01E0  00 6C 67 00 09 20 52 46  : 94
01E8  0C 11 00 56 67 00 09 16  : F9
01F0  0C 11 00 76 67 00 09 0E  : 11
01F8  0C 11 00 51 67 4A 0C 11  : 3C
-----------------------------------
SUM:  D3 8F 65 30 43 48 2A BA  42E9

0200  00 71 67 44 0C 01 00 28  : 51
0208  67 00 04 EE 0C 01 00 5E  : C4
0210  67 00 00 F8 0C 01 00 7E  : EA
0218  67 00 00 A8 0C 01 00 5F  : 7B
0220  67 00 00 A0 0C 01 00 3D  : 51
0228  67 5A 0C 01 00 5B 67 44  : D4
0230  14 C1 66 00 FF 6C 43 F9  : E2
0238  00 00 13 84 53 8A 95 C9  : D2
0240  23 CA 00 00 0E 1A 4E 75  : D8
0248  52 89 61 00 08 60 67 00  : 0B
0250  0A 46 61 00 09 58 17 43  : 6C
0258  00 03 66 00 FF 44 14 FC  : BC
0260  00 40 14 FC 00 4C 70 00  : 0C
0268  10 2B 00 01 61 00 08 E6  : 8B
0270  60 00 FF 2E 14 C1 12 19  : 8D
0278  14 C1 0C 01 00 5D 66 F6  : 9B
-----------------------------------
SUM:  1A 54 37 23 21 D6 0F 4F  1149

0280  60 00 FF 1E 61 00 08 26  : 0C
0288  67 0E 0C 19 00 30 56 EB  : 0B
0290  00 05 67 0E 60 00 FF 0A  : E3
0298  46 6B 00 05 67 04 60 00  : 81
02A0  FF 00 14 FC 00 79 70 30  : 28
02A8  D0 39 00 00 0D 68 61 00  : DF
02B0  08 A4 14 FC 00 2C 10 13  : 0B
02B8  E5 08 61 00 08 98 60 00  : 4E
02C0  FE E0 76 FF 61 00 07 E6  : A1
02C8  67 0A 61 00 08 E0 17 43  : 14
02D0  00 04 60 06 76 00 16 2B  : 21
02D8  00 04 78 00 18 2B 00 02  : C1
02E0  0C 01 00 7E 67 08 98 03  : 95
02E8  6A 0A 78 00 60 06 D8 03  : 2D
02F0  6A 02 78 7F 17 44 00 02  : C0
02F8  14 FC 00 40 14 FC 00 56  : B6
-----------------------------------
SUM:  22 5E 9A 84 26 32 A2 12  903F

0300  30 04 61 00 08 50 60 00  : 4D
0308  FE 98 61 00 07 A0 67 00  : 05
0310  09 6C 61 00 08 98 0C 03  : 85
0318  00 7E 62 00 09 60 3A 83  : 06
0320  52 89 61 00 07 88 67 00  : 32
0328  09 54 61 00 08 80 0C 03  : 55
0330  00 10 62 00 09 48 4A 03  : 10
0338  67 00 09 42 3B 7C 00 11  : 7A
0340  00 02 97 6D 00 02 52 89  : E3
0348  61 00 07 62 67 00 09 2E  : 68
0350  61 00 08 5A 0C 03 00 08  : DA
0358  62 00 09 22 4A 03 67 00  : 41
0360  09 1C 53 03 3B 43 00 04  : FD
0368  60 00 FE 36 23 C9 00 00  : 80
0370  0D 8A 23 C1 00 00 0D 8E  : 16
0378  7A 00 1A 2B 00 03 67 54  : 7D
-----------------------------------
SUM:  0D 1B EF B2 8E CB 00 42  1BFD

0380  61 00 01 DE 61 00 02 02  : A5
0388  4A 39 00 00 0D 67 6B 44  : A6
0390  9B 79 00 00 0E 02 6B 3C  : CB
0398  14 FC 00 40 14 FC 00 4C  : AC
03A0  30 39 00 00 0E 02 61 00  : DA
03A8  07 AC 10 01 E0 49 14 C1  : C2
03B0  0C 00 00 20 67 02 14 C0  : 69
03B8  14 FC 00 40 14 FC 00 4C  : AC
03C0  10 2B 00 03 61 00 07 8E  : 34
03C8  14 FC 00 52 61 00 05 94  : 5C
03D0  60 00 FD CE 22 79 00 00  : C6
03D8  0D 8A 22 39 00 00 0D 8E  : 8D
03E0  60 00 FE 4E 4A 2B 00 05  : 26
03E8  67 82 72 00 76 00 78 00  : 49
03F0  7A 00 7C 00 42 39 00 00  : 71
03F8  0D 69 61 00 01 64 30 01  : 6D
-----------------------------------
SUM:  90 2B 7D 29 E0 EF 22 51  BC8B

0400  61 00 06 58 54 04 2C 04  : 47
0408  3A 2D 00 00 38 2D 00 02  : CE
0410  61 00 01 76 26 05 86 C4  : 4D
0418  38 14 0C 44 FF FF 67 08  : 09
0420  42 79 00 00 0E 06 60 04  : 33
0428  78 00 18 13 70 00 10 13  : 36
0430  33 C0 00 00 0E 10 DB 79  : 65
0438  00 00 0E 10 33 C0 00 00  : 11
0440  0E 12 9B 79 00 00 0E 12  : 54
0448  61 00 05 18 30 39 00 00  : E7
0450  0E 06 67 14 14 FC 00 40  : DF
0458  14 FC 00 4C 61 00 06 F6  : B9
0460  61 00 01 C0 14 FC 00 26  : 58
0468  14 FC 00 40 14 FC 00 4C  : AC
0470  14 FC 00 31 14 FC 00 7C  : CD
0478  14 FC 00 3A 3A 39 00 00  : BD
-----------------------------------
SUM:  4F 82 41 91 8B 6D 78 98  5BC1

0480  0E 02 9A 79 00 00 0E 06  : 37
0488  6B 00 07 F2 34 2D 00 02  : C7
0490  E5 4A BA 42 65 2C 02 85  : 43
0498  00 00 FF FF 8A C2 30 05  : 7F
04A0  61 00 06 B2 2F 05 3A 02  : 89
04A8  53 85 61 72 14 FC 00 3A  : F5
04B0  14 FC 00 7C 2A 1F 48 45  : 62
04B8  4A 45 67 16 53 85 61 5E  : A3
04C0  60 10 14 FC 00 31 53 85  : 89
04C8  61 54 14 FC 00 3A 14 FC  : 0F
04D0  00 7C 4A 39 00 00 0D 67  : 73
04D8  66 3E 14 FC 00 79 14 FC  : 3D
04E0  00 38 14 FC 00 2C 70 00  : E4
04E8  10 39 00 00 0D 68 61 00  : 1F
04F0  06 64 38 BC FF FF 4A 39  : DF
04F8  00 00 0D 7B 66 00 04 34  : 26
-----------------------------------
SUM:  AD 05 07 C2 55 37 CA C2  A5AC

0500  14 FC 00 40 14 FC 00 4C  : AC
0508  10 2B 00 03 61 00 06 46  : EB
0510  14 FC 00 52 60 00 04 2E  : F4
0518  38 84 60 00 04 16 70 00  : A6
0520  10 2B 00 06 61 00 00 FC  : 9E
0528  3E 05 67 0A 02 07 00 07  : C4
0530  66 04 61 00 04 2E 61 00  : 5E
0538  01 B4 B8 79 00 00 0E 10  : 04
0540  6F 02 60 08 B8 79 00 00  : 0A
0548  0E 12 6C 0A 44 00 61 00  : 3B
0550  01 9C 61 00 01 98 51 CD  : B5
0558  FF CC 17 40 00 06 4E 75  : EB
0560  12 00 E1 49 12 3C 00 20  : AA
0568  14 11 0C 02 00 23 66 06  : C2
0570  14 3C 00 2B 60 0C 0C 02  : F5
0578  00 2B 67 06 0C 02 00 2D  : D3
-----------------------------------
SUM:  DC 83 78 EC BB CB 5B 6A  FAD2

0580  66 04 12 02 52 89 4E 75  : 1C
0588  61 00 05 22 67 22 61 00  : 72
0590  06 1C 0C 03 00 C0 62 00  : 53
0598  06 FA 4A 03 67 0C 20 3C  : 1C
05A0  00 00 00 C0 80 C3 36 00  : 39
05A8  60 10 36 3C 01 80 60 0A  : CD
05B0  16 2B 00 01 02 43 00 FF  : 86
05B8  30 03 61 56 61 14 33 C3  : 55
05C0  00 00 0E 02 E6 4B C6 ED  : F4
05C8  00 04 33 C3 00 00 0E 06  : 0E
05D0  4E 75 13 FC FF FF 00 00  : D0
05D8  0D 7B 61 00 05 00 0C 11  : 0B
05E0  00 26 57 F9 00 00 0D 67  : EA
05E8  66 04 52 89 60 22 33 C3  : BD
05F0  00 00 0D 8E 70 00 10 2B  : 46
05F8  00 03 67 14 96 40 67 0A  : C5
-----------------------------------
SUM:  3A 79 D6 62 54 BD 91 E0  CD8D

0600  5B F9 00 00 0D 7B 6B 02  : 49
0608  4E 75 36 39 00 00 0D 8E  : CD
0610  4E 75 0C 11 00 2E 66 08  : 7C
0618  E2 48 D6 40 52 89 60 F2  : 6D
0620  4E 75 48 E7 8A 00 74 00  : F0
0628  4A 44 6B 0E 0C 44 00 3F  : 96
0630  6F 14 04 44 00 3F 52 02  : 5E
0638  60 F2 4A 44 6A 08 06 44  : 9C
0640  00 3F 53 02 60 F4 14 FC  : F8
0648  00 79 70 30 D0 39 00 00  : 22
0650  0D 68 61 00 05 00 14 FC  : EB
0658  00 2C 10 04 E5 08 61 00  : 8E
0660  04 F4 DC 02 02 46 00 FF  : 1D
0668  10 39 00 00 0D 69 0C 06  : D1
0670  00 01 62 14 0C 00 00 3E  : C1
0678  67 28 14 FC 00 3E 13 FC  : EC
-----------------------------------
SUM:  C8 8C 9F 4F 94 DF B2 46  F656

0680  00 3E 00 00 0D 69 60 46  : 5A
0688  0C 06 00 0E 65 14 0C 00  : A5
0690  00 3C 67 0E 14 FC 00 3C  : FD
0698  13 FC 00 3C 00 00 0D 69  : C1
06A0  60 2C 4A 00 67 28 0C 00  : 71
06A8  00 3E 66 12 0C 06 00 01  : C9
06B0  63 1C 14 FC 00 3C 42 39  : 46
06B8  00 00 0D 69 60 10 0C 06  : F8
06C0  00 0E 64 0A 14 FC 00 3E  : CA
06C8  42 39 00 00 0D 69 DC 46  : 13
06D0  41 F9 00 00 0D E2 14 F0  : 2D
06D8  60 00 10 30 60 01 0C 00  : 0D
06E0  00 20 67 02 14 C0 4C DF  : 88
06E8  00 51 4E 75 4A 00 6A 04  : CC
06F0  98 43 60 02 D8 43 4E 75  : 1B
06F8  42 39 00 00 0D 7A 42 39  : 7D
-----------------------------------
SUM:  9F 2F C1 82 2A B8 15 30  B8B5

0700  00 00 0D 69 72 00 10 19  : 11
0708  61 00 03 88 67 00 05 88  : E0
0710  12 00 E1 49 12 3C 00 20  : AA
0718  14 19 0C 02 00 23 66 06  : CA
0720  14 3C 00 2B 60 0C 0C 02  : F5
0728  00 2B 67 06 0C 02 00 2D  : D3
0730  66 04 12 02 14 19 10 19  : D4
0738  61 00 03 58 67 00 05 58  : 80
0740  1A 00 E1 4D 1A 3C 00 20  : BE
0748  18 19 0C 04 00 23 66 06  : D0
0750  18 3C 00 2B 60 0C 0C 04  : FB
0758  00 2B 67 06 0C 04 00 2D  : D5
0760  66 04 1A 04 52 89 61 00  : C4
0768  03 44 67 1A 61 00 04 3E  : 6B
0770  0C 83 00 00 0C 00 62 00  : FD
0778  05 1A 0C 83 00 00 00 01  : AF
-----------------------------------
SUM:  26 E9 5A EA 17 7E D5 FD  03EC

0780  63 00 05 10 60 08 16 2B  : 21
0788  00 01 02 43 00 FF 33 C3  : 3B
0790  00 00 0E 02 61 00 03 46  : BA
```

```
0798  0C 11 00 26 57 F9 00 00  : 93
07A0  0D 67 66 02 52 89 30 01  : E8
07A8  61 00 02 B0 2C 04 30 05  : 78
07B0  61 00 02 A8 0C 02 00 3E  : 57
07B8  66 06 98 3C 00 0C 60 0A  : B6
07C0  0C 02 00 3C 66 04 D8 3C  : C8
07C8  00 0C 12 13 98 06 4A 04  : 1D
07D0  6B 04 74 01 60 04 44 04  : 90
07D8  74 FF 02 44 00 FF C8 FC  : 7C
07E0  00 3F D8 41 3A 04 9A 41  : 71
07E8  6A 02 44 05 33 FC 00 01  : E5
07F0  00 00 0E 0A 0C 43 00 0F  : 76
07F8  63 6A 30 03 02 40 00 01  : 43
---------------------------------
SUM:  5C 3B F9 F8 7B 2B D4 14  40A2

0800  66 18 0C 45 00 C0 64 12  : 05
0808  33 FC 00 02 00 00 0E 0A  : 49
0810  13 FC FF FF 00 00 0D 7A  : 94
0818  60 4A 20 03 02 80 00 00  : 4F
0820  FF FF 80 FC 00 03 02 80  : FF
0828  FF FF 00 00 66 18 0C 45  : CD
0830  00 80 64 12 33 FC 00 03  : 28
0838  00 00 0E 0A 13 FC FF FF  : 25
0840  00 00 0D 7A 60 1E 30 03  : 38
0848  02 40 00 03 66 16 0C 45  : 12
0850  00 40 64 10 33 FC 00 04  : E7
0858  00 00 0E 0A 13 FC FF FF  : 25
0860  00 00 0D 7A 61 00 00 FC  : E4
0868  16 01 4A 02 6A 06 53 06  : 2C
0870  D6 3C 00 3F 0C 03 00 3F  : 9F
0878  6F 06 96 3C 00 3F 52 06  : DE
---------------------------------
SUM:  67 9B 89 EF 91 C7 6C EF  8F92

0880  4A 06 6A 06 7C 0B 14 FC  : 57
0888  00 3E 0C 06 00 0B 6F 02  : CC
0890  7C 00 14 FC 00 40 14 FC  : DC
0898  00 4C 30 39 00 00 0E 0A  : CD
08A0  61 00 02 B2 30 39 00 00  : 7E
08A8  0E 02 53 40 3E 00 32 05  : 18
08B0  02 81 00 00 FF FF 82 C0  : C3
08B8  28 01 48 44 02 84 00 00  : 3B
08C0  FF FF E1 8C 88 C0 20 04  : D7
08C8  48 40 4A 40 67 02 52 04  : D1
08D0  C2 C2 3A 39 00 00 0E 0A  : 0F
08D8  70 00 53 05 66 0A 61 00  : 99
08E0  00 DA 3A 39 00 00 0E 0A  : 65
08E8  D0 04 64 02 D6 02 D6 01  : E9
08F0  3F 07 67 08 02 07 00 07  : C5
08F8  66 02 61 66 3E 1F 51 CF  : AC
---------------------------------
SUM:  4D FC 75 2A 56 06 6F BC  8B42

0900  FF DA 38 BC FF FF 4A 39  : 4E
0908  00 00 0D 67 66 24 4A 39  : 81
0910  00 00 0D 7A 67 04 53 8A  : CF
0918  60 18 14 FC 00 79 14 FC  : 11
0920  00 38 14 FC 00 2C 70 00  : E4
0928  10 39 00 00 0D 68 61 00  : 1F
0930  02 24 14 FC 00 40 14 FC  : 86
0938  00 4C 70 00 10 2B 00 01  : F8
0940  61 00 02 12 10 39 00 00  : BE
0948  0D 69 67 10 0C 00 00 3E  : 37
0950  67 06 14 FC 00 3E 60 04  : 1F
0958  14 FC 00 3C 61 0A 60 00  : 17
0960  F8 40 02 3C 00 FB 60 04  : D5
0968  00 3C 00 04 48 E7 FF FF  : 6D
0970  43 F9 00 00 13 84 67 0C  : 46
0978  20 4A 91 C9 B1 FC 00 00  : 71
---------------------------------
SUM:  B5 FD 0E F4 72 82 66 46  1D86

0980  00 60 65 30 42 12 95 C9  : A7
0988  23 CA 00 00 0E 1A 24 39  : 72
0990  00 00 0E 0C 4A 39 00 00  : 9D
0998  0D 7C 67 06 61 00 02 44  : 9D
09A0  60 06 72 06 70 F0 4E 4F  : DB
09A8  4C DF FF FF 45 F9 00 00  : 67
09B0  13 84 4E 75 4C DF FF FF  : 83
09B8  4E 75 2F 00 0C 03 00 3F  : 40
09C0  6F 06 96 3C 00 3F 52 06  : DE
09C8  4A 03 6A 06 D6 3C 00 3F  : 0E
09D0  53 06 4A 06 6A 06 7C 0B  : A0
09D8  14 FC 00 3E 0C 06 00 0B  : 6B
09E0  6F 06 7C 00 14 FC 00 3C  : 3D
09E8  14 FC 00 79 70 30 D0 39  : 32
09F0  00 00 0D 68 61 00 01 5E  : 35
09F8  14 FC 00 2C 0C 03 00 3F  : 8A
---------------------------------
SUM:  F4 8D 9B 4F 45 E6 A7 40  996B

0A00  66 24 14 FC 00 30 41 F9  : 04
0A08  00 00 0D C6 30 06 52 00  : 5B
0A10  D0 00 D1 C0 14 E8 00 00  : 5D
0A18  10 28 00 01 0C 00 00 20  : 65
0A20  67 28 14 C0 60 24 10 03  : FA
0A28  E5 08 61 00 01 28 41 F9  : B1
0A30  00 00 0D 92 30 06 D0 00  : A5
0A38  D1 C0 14 E8 00 00 10 28  : C5
0A40  00 01 0C 00 00 20 67 02  : 96
0A48  14 C0 4A 39 00 00 0D 7A  : DE
0A50  67 04 14 FC 00 26 20 1F  : E0
0A58  4E 75 23 C8 00 00 0D 8A  : 45
0A60  41 F9 00 00 0D 92 78 00  : 51
0A68  61 18 67 0E 41 F9 00 00  : 28
0A70  0D AC 78 00 61 0C 66 00  : 04
0A78  02 08 20 79 00 00 0D 8A  : 3A
---------------------------------
SUM:  DD 3B 14 41 90 4D 50 EC  31DC

0A80  4E 75 B0 58 67 0A 52 04  : 92
0A88  4A 10 66 F6 02 3C 00 FB  : EF
0A90  4E 75 02 00 00 DF 0C 00  : B0
0A98  00 41 65 0A 0C 00 00 47  : 03
0AA0  62 04 4A 00 4E 75 00 3C  : AF
0AA8  00 FF 4E 75 0C 11 00 30  : 0F
0AB0  65 0A 0C 11 00 39 62 04  : 2B
0AB8  4A 11 4E 75 00 3C 00 FF  : 59
0AC0  4E 75 14 C1 2F 09 61 00  : 31
0AC8  00 E4 20 3C 00 00 00 C0  : 00
0AD0  80 C3 17 40 00 01 22 5F  : 1C
0AD8  60 00 F6 C6 0C 11 00 20  : 59
0AE0  66 04 52 89 60 F6 4E 75  : 5E
0AE8  14 C1 2F 09 61 00 00 BE  : 2C
0AF0  43 F9 00 00 0D 6A 10 31  : F4
0AF8  30 00 17 40 00 02 22 5F  : 0A
---------------------------------
SUM:  12 33 48 28 D8 9D C3 B7  05DB

0B00  60 00 F6 9E 14 C1 14 D9  : B6
0B08  48 E7 01 40 61 00 00 9E  : 6F
0B10  17 83 60 00 4C DF 02 80  : A7
0B18  60 00 F6 86 14 FC 00 79  : 65
0B20  61 8A 66 06 76 00 16 13  : F6
0B28  60 0E 61 00 00 80 0C 03  : 5E
0B30  00 3F 62 00 01 62 16 83  : 9D
0B38  E5 0B 70 00 10 39 00 00  : A9
0B40  0D 68 D0 3C 00 30 61 0C  : 1E
0B48  14 FC 00 2C 20 03 61 04  : C4
0B50  60 00 F6 4E 48 E7 88 00  : 5B
0B58  42 39 00 00 0D 66 28 3C  : 52
0B60  00 00 27 10 02 80 00 00  : B9
0B68  FF FF 80 C4 4A 39 00 00  : C5
0B70  0D 66 66 0C 4A 00 67 0E  : A4
0B78  13 FC 00 FF 00 00 0D 66  : 81
---------------------------------
SUM:  A7 4A B9 FF 67 F0 34 C9  FF87

0B80  D0 3C 00 30 14 C0 48 40  : 98
0B88  02 80 00 00 FF FF 88 FC  : 04
0B90  00 0A 66 D6 4A 39 00 00  : C9
0B98  0D 66 67 06 4C DF 00 11  : 1C
0BA0  4E 75 14 FC 00 30 4C DF  : 2E
0BA8  00 11 4E 75 2F 00 70 00  : 73
0BB0  76 00 10 19 90 3C 00 30  : 9B
0BB8  6B 0E 0C 00 00 09 62 08  : F8
0BC0  C6 FC 00 0A D6 40 60 EA  : 2C
0BC8  20 1F 53 89 4E 75 4A 2F  : 57
0BD0  00 0F 56 F9 00 00 0D 7C  : E7
0BD8  42 80 61 00 00 BE 42 80  : A3
0BE0  4E 75 4A 39 00 00 0D 7D  : D0
0BE8  66 1E 3F 3C 00 20 48 79  : E0
0BF0  00 00 0D 7E FF 3C 5C 8F  : B1
0BF8  33 C0 00 00 0E 18 6B 6E  : F2
---------------------------------
SUM:  1D BD EB 15 99 33 03 6C  1CAF

0C00  13 FC FF FF 00 00 0D 7D  : 97
0C08  45 F9 00 00 13 7C 14 FC  : DD
0C10  00 28 14 FC 00 74 30 02  : DE
0C18  61 00 FF 3A 14 FC 00 29  : D3
0C20  95 FC 00 00 13 7C 48 52  : BA
0C28  48 79 00 00 13 7C 3F 39  : C8
0C30  00 00 0E 18 FF 40 4F EF  : A3
0C38  00 0A 4A 80 6B 30 2F 39  : D7
0C40  00 00 0E 1A 48 79 00 00  : E9
0C48  13 84 3F 39 00 00 0E 18  : 35
0C50  FF 40 4F EF 00 0A 4A 80  : 51
0C58  6B 14 70 0D 61 02 70 0A  : D9
0C60  3F 39 00 00 0E 18 3F 00  : DD
0C68  FF 1D 58 8F 4E 75 FF 1F  : E4
0C70  70 37 60 3E 70 36 60 3A  : 85
0C78  70 35 60 36 70 34 60 32  : 71
---------------------------------
SUM:  31 36 8E 1F 9C D0 1C 84  6296

0C80  13 C0 00 00 12 3D E0 48  : 4A
0C88  13 C0 00 00 12 3C 70 33  : C4
0C90  60 20 70 32 60 1C 70 31  : 3F
0C98  60 18 41 FA 01 C2 21 40  : D7
0CA0  00 06 4E 75 4A B9 00 00  : CC
0CA8  0E 34 6A 04 70 09 60 02  : 8B
0CB0  4E 75 2E 79 00 00 0E 14  : 8C
0CB8  2F 00 53 80 E5 80 22 7B  : 04
0CC0  08 0A 70 FF 61 D4 20 1F  : F5
0CC8  4E 75 00 00 0E 38 00 00  : 09
0CD0  0E 4F 00 00 0E 66 00 00  : D1
0CD8  0E 85 00 00 0E 9C 00 00  : 3D
0CE0  0E B3 00 00 0E CC 00 00  : 9B
0CE8  10 DE 00 00 0E E7 00 00  : E3
0CF0  00 00 00 00 00 00 00 00  : 00
0CF8  00 00 00 00 00 00 00 00  : 00
---------------------------------
SUM:  01 4B 5A 9D CB 5A 91 9C  F19D

0D00  00 00 00 00 00 00 00 00  : 00
0D08  00 00 00 00 00 00 00 00  : 00
0D10  00 00 00 00 0E FC 00 00  : 0A
0D18  0F 19 00 00 0F 38 00 00  : 6F
0D20  0F 47 00 00 0F 5E 00 00  : C3
0D28  0F 7D 00 00 0F 9A 00 00  : 35
0D30  0F B7 00 00 0F D2 00 00  : A7
0D38  0F E5 00 00 0F FE 00 00  : 01
0D40  10 19 00 00 10 2E 00 00  : 67
0D48  10 41 00 00 10 58 00 00  : B9
0D50  10 6D 00 00 10 82 00 00  : 0F
0D58  10 97 00 00 10 AA 00 00  : 61
0D60  10 C5 00 00 10 DE 00 00  : C3
0D68  10 F1 00 00 11 00 00 00  : 12
0D70  11 1B 00 00 11 36 00 00  : 73
0D78  11 4F 00 00 11 6E 00 00  : DF
---------------------------------
SUM:  CD F7 00 00 DC 30 00 00  A6CC

0D80  11 87 00 00 11 A2 00 00  : 4B
0D88  11 BB 00 00 11 D6 00 00  : B3
0D90  11 F7 00 00 12 22 00 00  : 3C
0D98  12 52 00 00 12 77 00 00  : ED
0DA0  12 9D 00 00 12 D0 00 00  : 91
0DA8  00 00 55 57 5A 5D 5F 62  : 24
0DB0  65 67 6A 6D 6F 72 75 77  : 70
0DB8  7A 7D 00 00 00 00 5A 4D  : 9E
0DC0  55 53 49 43 2E 54 4D 50  : 53
0DC8  00 00 00 00 00 00 00 00  : 00
0DD0  00 00 43 20 43 2B 44 20  : 35
0DD8  44 2B 45 20 46 20 46 2B  : AB
0DE0  47 20 47 2B 41 20 41 2B  : A6
0DE8  42 20 00 00 43 20 44 2D  : 36
0DF0  44 20 45 2D 45 20 46 20  : A1
0DF8  47 2D 47 20 41 2D 41 20  : AA
---------------------------------
SUM:  E3 17 63 BF E2 DC 11 59  C5DE

0E00  42 2D 42 20 00 00 43 20  : 34
0E08  43 2B 44 20 44 2B 45 20  : A6
0E10  46 20 46 2B 47 20 47 2B  : B0
0E18  41 20 41 2B 42 20 3C 43  : AE
0E20  00 00 42 2D 42 20 43 20  : 34
0E28  43 2B 44 20 44 2B 45 20  : A6
0E30  46 20 46 2B 47 20 47 2B  : B0
0E38  41 20 41 2B 42 20 43 20  : 92
0E40  43 2B 00 00 00 00 00 00  : 6E
0E48  00 00 00 00 00 00 00 00  : 00
0E50  00 00 00 00 00 00 00 00  : 00
0E58  00 00 00 00 00 00 00 00  : 00
0E60  00 00 00 00 00 00 00 00  : 00
0E68  00 00 00 00 00 00 00 00  : 00
0E70  00 00 00 00 00 00 00 00  : 00
0E78  83 65 83 93 83 7C 82 CC  : 4B
---------------------------------
SUM:  9C 93 9D CC 5F 72 9F 05  5B73

0E80  8E 77 92 E8 82 AA 96 B3  : F4
0E88  8C F8 82 C5 82 B7 00 83  : 87
0E90  67 83 89 83 62 83 4E 94  : BD
0E98  D4 8D 86 82 AA 96 B3 8C  : E8
0EA0  F8 82 C5 82 B7 00 82 8D  : 87
0EA8  82 95 82 93 82 89 82 83  : 3C
0EB0  81 40 82 83 82 8D 82 84  : DB
0EB8  81 40 82 85 82 92 82 92  : F0
0EC0  82 8F 82 92 00 83 54 83  : 7F
0EC8  43 83 59 82 CC 8E 77 92  : 04
0ED0  E8 82 AA 96 B3 8C F8 82  : 63
0ED8  C5 82 B7 00 83 81 83 82  : 07
0EE0  83 8A 81 5B 8A 6D 95 DB  : 50
0EE8  82 C5 82 AB 82 DC 82 B9  : 0D
0EF0  82 F1 00 83 60 83 83 83  : DF
0EF8  93 83 6C 83 8B 94 D4 8D  : 85
---------------------------------
SUM:  5D EF 19 85 46 A0 53 39  3214

0F00  86 82 AA 96 B3 8C F8 82  : 01
0F08  C5 82 B7 00 94 7A 97 F1  : 94
0F10  82 CC 8E 77 92 E8 82 C9  : 18
0F18  8C EB 82 E8 82 AA 82 A0  : 2F
0F20  82 E8 82 DC 82 B7 00 89  : 8A
0F28  B9 8A 79 97 70 8A D6 90  : B3
0F30  94 82 CD 96 B3 8C F8 82  : 32
0F38  C5 82 B7 00 82 6C 82 6C  : DA
0F40  82 6B 82 CC 95 B6 96 40  : 5C
```


▶「ギュイィィィィ……ン」と見事にポルタメントのかかった音は，ハードディスクがクラッシュしたときの音です（これで2台目)。　溝口　信太郎（20）愛知県

```
0F48  82 C9 8C EB 82 E8 82 AA  : 58
0F50  82 A0 82 E8 82 DC 82 B7  : 23
0F58  00 81 6D 81 6E 82 CC 92  : BD
0F60  86 82 CC 8E 77 92 E8 82  : D5
0F68  C9 8C EB 82 E8 82 AA 82  : 58
0F70  A0 82 E8 82 DC 82 B7 00  : A1
0F78  81 6E 82 AA 82 A0 82 E8  : A7
-----------------------------------
SUM:  E3 84 0E 5A 46 03 14 02  1207

0F80  82 DC 82 B9 82 F1 00 8C  : 98
0F88  4A 82 E8 95 D4 82 B5 82  : D6
0F90  CC 92 6C 82 AA 96 B3 8C  : CB
0F98  F8 82 C5 82 B7 00 8C 4A  : 4E
0FA0  82 E8 95 D4 82 B5 94 D4  : 72
0FA8  8D 86 82 CC 8E 77 92 E8  : E0
0FB0  82 AA 82 A0 82 E8 82 DC  : 16
0FB8  82 B9 82 F1 00 8C 4A 82  : 06
0FC0  E8 95 D4 82 B5 94 D4 8D  : 7D
0FC8  86 82 CC 8E 77 92 E8 82  : D5
0FD0  AA 96 B3 8C F8 82 C5 82  : 40
0FD8  B7 00 83 49 83 4E 83 5E  : 35
0FE0  81 5B 83 75 82 CC 94 D4  : 8A
0FE8  8D 86 82 AA 82 A0 82 E8  : CB
0FF0  82 DC 82 B9 82 F1 00 83  : 8F
0FF8  49 83 4E 83 5E 81 5B 83  : 5A
-----------------------------------
SUM:  4B 30 61 C3 D4 7D 5B AF  BDFF

1000  75 82 CC 8E 77 92 E8 82  : C4
1008  AA 96 B3 8C F8 82 C5 82  : 40
1010  B7 00 92 B7 82 B3 82 CC  : 83
1018  92 6C 82 AA 96 B3 8C F8  : F7
1020  82 C5 82 B7 00 83 67 83  : ED
1028  89 83 62 83 4E 97 65 97  : D2
1030  CA 82 AA 91 AB 82 E8 82  : 1E
1038  DC 82 B9 82 F1 00 89 B9  : CC
1040  90 46 94 D4 8D 86 82 CC  : 9F
1048  8E 77 92 E8 82 AA 82 A0  : CD
1050  82 E8 82 DC 82 B9 82 F1  : 76
1058  00 81 97 82 76 82 CC 92  : F0
1060  6C 82 AA 82 A0 82 E8 82  : A6
1068  DC 82 B9 82 F1 00 81 97  : A2
1070  82 76 82 CC 92 6C 82 AA  : 70
1078  96 B3 8C F8 82 C5 82 B7  : 4D
-----------------------------------
SUM:  19 23 8A AA 1D 34 B7 86  B52A

1080  00 83 65 83 93 83 7C 82  : 7F
1088  CC 92 6C 82 AA 82 A0 82  : 9A
1090  E8 82 DC 82 B9 82 F1 00  : F4
1098  83 65 83 93 83 7C 82 CC  : 4B
10A0  92 6C 82 AA 96 B3 8C F8  : F7
10A8  82 C5 82 B7 00 92 B7 82  : 4B
10B0  B3 82 CC 92 6C 82 AA 82  : AD
10B8  A0 82 E8 82 DC 82 B9 82  : 25
10C0  F1 00 89 B9 97 CA 82 CC  : E2
10C8  92 6C 82 AA 82 A0 82 E8  : B6
10D0  82 DC 82 B9 82 F1 00 89  : 95
10D8  B9 97 CA 82 CC 92 6C 82  : E8
10E0  AA 96 B3 8C F8 82 C5 82  : 40
10E8  B7 00 83 4C 81 5B 83 52  : 37
10F0  81 5B 83 68 82 CC 92 6C  : 13
10F8  82 AA 82 A0 82 E8 82 DC  : 16
-----------------------------------
SUM:  C0 AB 7A 0D 3B CA 01 29  D865

1100  82 B9 82 F1 00 83 4C 81  : FE
1108  5B 83 52 81 5B 83 68 82  : 79
1110  CC 92 6C 82 AA 96 B3 8C  : CB
1118  F8 82 C5 82 B7 00 89 B9  : BA
1120  90 46 94 D4 8D 86 82 AA  : 7D
1128  96 B3 8C F8 82 C5 82 B7  : 4D
1130  00 81 70 82 AA 82 A0 82  : C1
1138  E8 82 DC 82 B9 82 F1 00  : F4
1140  81 6F 81 70 82 CC 92 86  : 47
1148  82 C9 89 B9 95 84 82 AA  : D2
1150  91 BD 82 B7 82 AC 82 DC  : 13
1158  82 B7 00 81 6F 81 70 82  : 9C
1160  CC 92 86 82 C9 89 B9 95  : 06
```

```
1168  84 82 AA 82 A0 82 E8 82  : BE
1170  DC 82 B9 82 F1 00 82 70  : 7C
1178  82 CC 8E 77 92 E8 82 C9  : 18
-----------------------------------
SUM:  73 5A 74 A4 22 5B 30 09  47C1

1180  8C EB 82 E8 82 AA 82 A0  : 2F
1188  82 E8 82 DC 82 B7 00 81  : 82
1190  6F 81 70 82 CC 92 86 82  : 48
1198  CC 95 B6 96 40 82 C9 8C  : C4
11A0  EB 82 E8 82 AA 82 A0 82  : 25
11A8  E8 82 DC 82 B7 00 82 78  : 79
11B0  82 CC 8E 77 92 E8 82 C9  : 18
11B8  8C EB 82 E8 82 AA 82 A0  : 2F
11C0  82 E8 82 DC 82 B7 00 81  : 82
11C8  97 82 6B 82 CC 8E 77 92  : 69
11D0  E8 82 C9 8C EB 82 E8 82  : 96
11D8  AA 82 A0 82 E8 82 DC 82  : 16
11E0  B7 00 82 6F 82 CC 8E 77  : FB
11E8  92 E8 82 C9 8C EB 82 E8  : A6
11F0  82 AA 82 A0 82 E8 82 DC  : 16
11F8  82 B7 00 83 5E 83 43 82  : 62
-----------------------------------
SUM:  22 5B DA 06 94 F4 07 66  3420

1200  CC 8E 77 92 E8 82 C9 8C  : 22
1208  EB 82 E8 82 AA 82 A0 82  : 25
1210  E8 82 DC 82 B7 00 CF B8  : 06
1218  DB BA CF DD C4 DE 82 CC  : 31
1220  8E 67 97 70 96 40 82 C9  : 1D
1228  8C EB 82 E8 82 AA 82 A0  : 2F
1230  82 E8 82 DC 82 B7 00 CF  : D0
1238  B8 DB BA CF DD C4 DE 82  : 1D
1240  C9 91 CE 82 B5 82 C4 82  : 27
1248  CC 89 B9 92 B7 8E 77 92  : EE
1250  E8 82 C9 8C EB 82 E8 82  : 96
1258  AA 82 A0 82 E8 82 DC 82  : 16
1260  B7 00 CF B8 DB BA CF DD  : 7F
1268  C4 DE 92 86 82 C9 8C EB  : 7C
1270  82 C1 82 BD B7 B0 BA B0  : 53
1278  C4 DE 20 28 20 20 29 82  : D5
-----------------------------------
SUM:  B6 FC 52 BB F7 AE D9 5E  4F8D

1280  AA 97 70 82 A2 82 E7 82  : C0
1288  EA 82 C4 82 A2 82 DC 82  : 34
1290  B7 00 D3 BC DE AD DA B0  : 5B
1298  BC AE DD A5 CF B8 DB BA  : 08
12A0  CF DD C4 DE 92 86 82 C9  : B1
12A8  8C EB 82 E8 82 AA 82 A0  : 2F
12B0  82 E8 82 DC 82 B7 00 91  : 92
12B8  8A 91 CE CE DE D8 AD B0  : CA
12C0  D1 A5 CF B8 DB BA CF DD  : 3E
12C8  C4 DE 92 86 82 C9 8C EB  : 7C
12D0  82 E8 82 AA 82 A0 82 E8  : 22
12D8  82 DC 82 B7 00 8C 4A 82  : EF
12E0  E8 95 D4 82 B5 89 F1 90  : 92
12E8  94 82 AA 91 E5 82 AB 82  : E5
12F0  A2 82 A9 81 41 8C 4A 82  : E7
12F8  E8 95 D4 82 B5 82 CC 94  : 6A
-----------------------------------
SUM:  0D 7D DA 8A D4 F0 02 72  8AFC

1300  CD 88 CD 82 AA 8D 4C 82  : A9
1308  B7 82 AC 82 DC 82 B7 00  : 7C
1310  83 74 83 40 83 43 83 8B  : 8E
1318  8D EC 90 AC 82 C9 8E B8  : 46
1320  94 73 82 B5 82 DC 82 B5  : D3
1328  82 BD 00 00 00 00 00 00  : 3F
1330  00 00 00 00 00 00 00 00  : 00
1338  00 00 00 00 00 00 00 00  : 00
1340  00 00 00 00 00 00 00 00  : 00
1348  00 00 00 00 00 00 00 00  : 00
1350  00 00 00 00 00 00 00 00  : 00
1358  00 00 00 00 00 00 00 00  : 00
1360  00 00 00 00 00 00 00 00  : 00
1368  00 00 00 00 00 00 00 00  : 00
1370  00 00 00 00 00 00 00 00  : 00
1378  00 00 00 00 00 00 00 00  : 00
-----------------------------------
```

```
SUM:  AA 9A 0E A5 0D F7 96 7A  17F3
```

1380_H～$177F_H$は0で埋める

```
1780  00 00 00 00 00 00 00 00  : 00
1788  00 00 00 00 00 00 00 00  : 00
1790  00 00 00 00 00 00 00 00  : 00
1798  00 00 00 00 00 00 00 00  : 00
17A0  00 00 00 00 00 00 00 00  : 00
17A8  00 00 00 00 00 00 00 00  : 00
17B0  00 00 00 00 00 00 00 00  : 00
17B8  00 00 00 00 00 00 00 00  : 00
17C0  00 00 00 00 00 00 00 04  : 04
17C8  00 04 00 04 00 04 00 04  : 10
17D0  00 04 00 04 00 04 00 04  : 10
17D8  00 04 00 2A 00 04 00 0E  : 40
17E0  00 04 00 06 00 2A 00 08  : 3C
17E8  00 08 00 06 00 06 00 28  : 3C
17F0  00 0A 00 12 00 08 00 06  : 2A
17F8  00 24 00 06 00 06 00 06  : 36
-----------------------------------
SUM:  00 46 00 56 00 4A 00 56  9170

1800  00 10 00 AC 00 0A 00 68  : 2E
1808  00 C4 00 06 00 16 00 08  : E8
1810  00 10 00 34 00 06 00 1A  : 64
1818  00 2C 00 10 00 06 00 06  : 48
1820  00 06 00 0A 00 30 00 06  : 46
1828  00 50 00 16 00 0E 00 44  : B8
1830  00 0A 00 7A 00 0C 00 0A  : 9A
1838  00 0E 00 0C 00 12 00 0A  : 36
1840  00 42 00 1C 00 18 00 1A  : 90
1848  00 1C 00 12 00 08 00 28  : 5E
1850  00 06 00 90 00 0E 00 52  : F6
1858  00 1C 00 08 00 24 00 08  : 50
1860  00 18 00 08 00 3C 00 0A  : 66
1868  00 2E 00 10 00 24 00 08  : 6A
1870  00 1A 00 1C 00 2C 00 18  : 7A
1878  00 06 00 06 00 18 00 42  : 66
-----------------------------------
SUM:  00 64 00 9C 00 7E 00 F6  E77C

1880  00 18 00 28 00 1C 00 10  : 6C
1888  00 06 00 0C 00 0E 00 76  : 96
1890  00 4C 00 1C 00 14 00 0E  : 8A
1898  00 1A 00 3E 00 10 00 0C  : 74
18A0  00 0A 00 0A 00 06 00 18  : 32
18A8  00 08 00 06 00 10 00 06  : 24
18B0  00 06 00 16 00 20 00 08  : 44
18B8  00 1C 00 0E 00 16 00 04  : 44
18C0  00 04 00 04 00 04 00 04  : 10
18C8  00 04 00 04 00 04 00 28  : 34
18D0  00 04 00 04 00 04 00 04  : 10
18D8  00 04 00 04 00 04 00 04  : 10
18E0  00 04 00 04 00 04 00 04  : 10
18E8  00 04 00 04 00 04 00 04  : 10
18F0  00 04 00 04 00 04 00 04  : 10
18F8  00 04 00 04 00 04 00 04  : 10
-----------------------------------
SUM:  00 D8 00 E2 00 BA 00 0E  963C

1900  00 04 00 04 00 04 00 04  : 10
1908  00 04 00 04 00 04 00 04  : 10
1910  00 04 00 04 00 04 00 04  : 10
1918  00 00 00 00 00 00 00 00  : 00
1920  00 00 00 00 00 00 00 00  : 00
1928  00 00 00 00 00 00 00 00  : 00
1930  00 00 00 00 00 00 00 00  : 00
1938  00 00 00 00 00 00 00 00  : 00
1940  00 00 00 00 00 00 00 00  : 00
1948  00 00 00 00 00 00 00 00  : 00
1950  00 00 00 00 00 00 00 00  : 00
1958  00 00 00 00 00 00 00 00  : 00
1960  00 00 00 00 00 00 00 00  : 00
1968  00 00 00 00 00 00 00 00  : 00
1970  00 00 00 00 00 00 00 00  : 00
1978  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  00 0C 00 0C 00 0C 00 0C  ADEF
```

リスト6　ZMUSIC.Sソースリスト

```
 1: ***********************************************
 2: *                                             *
 3: *               Z M U S I C . F N C           *
 4: *                                             *
 5: *           PROGRAMMED BY  Z.NISHIKAWA        *
 6: *                                             *
 7: ***********************************************
 8:
 9:         .include        iocscall.mac
10:         .include        doscall.mac
11:         .text
12:
13: max:    equ     3072    *最長音長 16*192 16小節分
14: exp:    equ     15      *経験的処理の目安
15:
16: nl:     equ     15      *振幅
17: spd:    equ     13      *speed
18: delay:  equu    1       *delay
19: line_max:       equ     96      *文字列がどれ位長くなったら
20:                                 *ひとまず送るか。目安(96-200)
21:
22: ****************************************
23: *       インフォメーションテーブル     *
24: ****************************************
25:
26: info_tbl:
27:             .dc.l   init
28:             .dc.l   run
29:             .dc.l   end
30:             .dc.l   system
31:             .dc.l   break
32:             .dc.l   ctrl_d
33:             .dc.l   yobi
34:             .dc.l   yobi
```

▶カードプログラムがたくさん出てきますね。とってもうれしいです。高校野球も終わって，プロ野球もほぼセ・パともに決まってしまいました。編集部の皆さんは，どこのファンが多いですか？　ちなみに僕は巨人です。　　笹山　克己（16）石川県

```
 35:         .dc.l   token_tbl
 36:         .dc.l   param
 37:         .dc.l   exec_adr
 38:         .ds.b   20
 39:
 40: token_tbl:
 41:         .dc.b   'm_trk2',$00    *外部関数名
 42:         .dc.b   'm_switch',$00
 43:         .dc.b   $00
 44:         .even
 45: param:
 46:         .dc.l   mtrk2_p
 47:         .dc.l   msw_p
 48: mtrk2_p:
 49:         .dc.w   $0004   *1バイト符号無し整数
 50:         .dc.w   $0008   *文字列
 51:         .dc.w   $8001   *4バイト符号付き整数(戻り値)
 52: msw_p:
 53:         .dc.w   $0004
 54:         .dc.w   $8001
 55:         .even
 56: exec_adr:
 57:         .dc.l   mtrk
 58:         .dc.l   m_switch
 59: init:
 60:         clr.b   out_flg
 61:         move.l  #$000001f0,d1
 62:         suba.l  a1,a1
 63:         IOCS    _B_INTVCS
 64:         movea.l d0,a1           *a1=設定前の処理アドレス
 65:         move.l  #$000001f0,d1
 66:         IOCS    _B_INTVCS
 67:         cmpa.l  #$00fe0000,a1
 68:         bcs     clr_err_flg
 69:         move.l  #$ffffffff,err_flg      *opmdrv未登録です...
 70:         rts
 71:
 72: clr_err_flg:
 73:         clr.l   err_flg
 74:         rts
 75: run:
 76: ctrl_d:
 77:         lea     ch_work,a0      *ﾁｬﾝﾈﾙﾜｰｸ
 78:         lea     lfo_wk,a3       *lfo work
 79:         lea     last_kf,a4      *last key fraction
 80:
 81:         moveq.l #8-1,d0
 82: init_lp:
 83:         move.l  #$00306a00,(a0)+  *set default kf,@L,@v,@q
 84:         move.l  #$03000100,(a0)+  *set default _~,lfosw,sgn,dmy
 85:         move.l  #$000f0003,(a3)+  *set default lfo parameters nl,spd,dly
 86:         move.l  #$00030000,(a3)+
 87:         move.w  #-1,(a4)+       *last_kf_flg
 88:         dbra    d0,init_lp
 89:         clr.b   opn_flg
 90:         rts
 91:
 92: end:
 93: break:
 94: system:
 95: yobi:
 96:         rts
 97:
 98: mtrk:
 99:         move.l  sp,svsp         *ｽﾀｯｸをsave
100:         bsr     err?
101:         movea.l $0016(a7),a1    *文字列のスタートアドレス
102:         move.l  $000c(a7),d2    *トラック番号
103:
104:         move.l  d2,trk_num
105:         bsr     com_chk         *マクロコマンド処理へ…
106:         move.l  trk_num,d2
107:
108:         tst.b   out_flg
109:         beq.s   opmout
110:         bsr     file_out
111:         bra.s   ext_mtrk2
112: opmout:
113:         moveq.l #$06,d1
114:         IOCS    _OPMDRV
115:         tst.l   d0
116:         bne     err_msg_sel
117: ext_mtrk2:
118:         clr.l   d0
119:         bsr     junbi
120:         clr.l   d0
121:         rts
122:
123: com_chk:
124:         * <     a1=文字列スタートアドレス
125:         * <     d2=トラック番号
126:
127:         lea     buffer,a2       *buffer 先頭アドレス
128:         lea     ch_work,a3      *チャンネルワーク
129:         lea     last_kf,a4      *last key fraction
130:         lea     lfo_wk,a5       *lfo parameters
131:
132:         moveq.l #0,d7
133:         move.b  d2,d7
134:         subq.b  #1,d7
135:         andi.b  #7,d7
136:         move.b  d7,ch_num
137:
138:         move.l  d7,d0
139:         lsl.b   d0
140:         adda.l  d0,a4           *a4=last kf wk+2*ch_n
141:
142:         move.b  d7,d0
143:         lsl.w   #parm2,d0       *d0=ch_num*parm
144:         adda.l  d0,a5           *a0=lfoのﾊﾟﾗﾒｰﾀ
145:
146:         lsl.b   #wk_sz2,d7      *d7=d7*2^wk_sz2
147:                                 *d7=ﾜｰｸｾｯﾄ用のﾎﾟｲﾝﾀ
148:         adda.l  d7,a3
149: loop_chk:
150:         move.b  (a1)+,d1
151:         move.b  d1,d0           *d0=d1
152:
153:         andi.b  #$df,d0         *大文字小文字変換
154:
155:         cmpi.b  #'A',d0
156:         bcs.s   chk_mcom
157:         cmpi.b  #'G',d0
158:         bls     modulation
159:
160: chk_mcom:
161:         cmpi.b  #'K',d0         *Kコマンド
162:         beq     detune
163:
164:         cmpi.b  #'L',d0         *Lの値をﾜｰｸにセット
165:         beq     l_wk_set1
166:
167:         cmpi.b  #'V',d0
168:         beq     v_wk_set1       *Vの値をﾜｰｸにｾｯﾄ
169:
170:         moveq.l #1,d6           *d6=1
171:         cmpi.b  #'@',d1         *d1であることに注意!!
172:         bne.s   not@
173:         cmpi.b  #'L',(a1)
174:         beq     wk_set2         *@Lの値をﾜｰｸにｾｯﾄ
175:         cmpi.b  #'l',(a1)
176:         beq     wk_set2         *       "
177:         addq.w  #1,d6           *d6=2
178:         cmpi.b  #'V',(a1)
179:         beq     wk_set2         *@Vの値をﾜｰｸにｾｯﾄ
180:         cmpi.b  #'v',(a1)
181:         beq     wk_set2
182:         cmpi.b  #'Q',(a1)
183:         beq     atq_set
184:         cmpi.b  #'q',(a1)
185:         beq     atq_set
186: not@:
187:         cmpi.b  #'(',d1         *d1であることに注意!!
188:         beq     portament       *ﾎﾟﾙﾀﾒﾝﾄ
189:
190:         cmpi.b  #'^',d1
191:         beq     mod_p_set       *ﾓｼﾞｭﾚｰｼｮﾝ ﾊﾟﾗﾒｰﾀ ｾｯﾄ
192:
193:         cmpi.b  #'~',d1
194:         beq     v_up
195:
196:         cmpi.b  #'_',d1
197:         beq     v_dwn
198:
199:         cmpi.b  #'=',d1
200:         beq     set_lfosw       *set lfo switch
201:
202:         cmpi.b  #'[',d1         *分岐ｺﾏﾝﾄﾞはｽｷｯﾌﾟ
203:         beq     skip_com
204:
205: just_wrt:
206:         move.b  d1,(a2)+        *ﾏｸﾛｺﾏﾝﾄﾞがない場合は単にｺﾋﾟｰ
207:         bne     loop_chk
208:
209:         lea     buffer,a1       *a1=buffer
210:
211:         subq.l  #1,a2
212:         suba.l  a1,a2
213:         move.l  a2,mj_lng
214:         rts
215:
216: atq_set:
217:         addq.l  #1,a1
218:         bsr     num_chk
219:         beq     err_in_macro
220:         bsr     num_get
221:
222:         move.b  d3,3(a3)        *save default value
223:         bne     loop_chk
224:
225:         move.b  #'@',(a2)+
226:         move.b  #'L',(a2)+
227:         moveq.l #0,d0
228:         move.b  1(a3),d0
229:         bsr     num_wrt         *write default length
230:         bra     loop_chk
231:
232: skip_com:
233:         move.b  d1,(a2)+        *write '['
234: skip_com_lp:
235:         move.b  (a1)+,d1
236:         move.b  d1,(a2)+
237:         cmpi.b  #']',d1
238:         bne.s   skip_com_lp
239:         bra     loop_chk
240:
241: set_lfosw:
242:         bsr     num_chk
243:         beq.s   rev_sw          *=のみならｽｲｯﾁ反転
244:         cmpi.b  #'0',(a1)+
245:         sne.b   5(a3)           *1以上ならon,0ならoff
246:         beq.s   _off
247:         bra     loop_chk
248: rev_sw:
249:         not     5(a3)           *reverse lfo_sw
250:         beq.s   _off
251:         bra     loop_chk
252: _off:
253:         move.b  #'y',(a2)+
254:         moveq.l #48,d0
255:         add.b   ch_num,d0
256:         bsr     num_wrt
257:
258:         move.b  #',',(a2)+
259:
260:         move.b  (a3),d0 *default kf
261:         lsl.b   #2,d0           *y=y*4 (0-252)
262:         bsr     num_wrt
263:         bra     loop_chk
264: v_up:
265: v_dwn:
266:         moveq.l #-1,d3
267:
268:         bsr     num_chk
269:         beq.s   vud1
270:         bsr     num_get         *d3=value
271:         move.b  d3,4(a3)
272:         bra.s   kasan_v
273: vud1:
274:         moveq.l #0,d3
275:         move.b  4(a3),d3
276: kasan_v:
277:         moveq.l #0,d4
278:         move.b  2(a3),d4        *default @v
279:         cmpi.b  #'~',d1         *which ?
280:         beq.s   _up
281: *dwn
282:         sub.b   d3,d4
283:         bpl.s   wrt_v
284:         moveq.l #0,d4
285:         bra.s   wrt_v
286: _up:
287:         add.b   d3,d4
288:         bpl.s   wrt_v
289:         moveq.l #127,d4
290: wrt_v:
291:         move.b  d4,2(a3)        *save default @v
292:
```

▶X68000が届いた。黒いボディが6畳1間の2人部屋に映えるぜ。しかしうちの部屋は最近，ゲームセンターになりつつあるのであった。　信太　徹（20）神奈川県

```
293:         move.b  #'@',(a2)+
294:         move.b  #'V',(a2)+      *write '@V'
295:         move.w  d4,d0
296:         bsr     num_wrt
297:
298:         bra.w   loop_chk
299:
300: mod_p_set:
301:         bsr     num_chk
302:         beq     mod_err         *数字じゃない
303:         bsr     num_get         *d3=width
304:         cmpi.b  #126,d3
305:         bhi     mod_err
306:
307:         move.w  d3,(a5)         *save 振幅
308:         addq.l  #1,a1           *skip ','
309:
310:         bsr     num_chk
311:         beq     mod_err
312:         bsr     num_get         *d3=spd
313:         cmpi.b  #16,d3
314:         bhi     mod_err
315:         tst.b   d3
316:         beq     mod_err         *spdが(1-16)の範囲でなければerr!
317:
318:         move.w  #17,2(a5)
319:         sub.w   d3,2(a5)        *d4=17-spd
320:         addq.l  #1,a1           *skip ','
321:
322:         bsr     num_chk
323:         beq     mod_err
324:         bsr     num_get         *d3=delay
325:         cmpi.b  #8,d3
326:         bhi     mod_err         *too big
327:         tst.b   d3
328:         beq     mod_err         *too small
329:         subq.b  #1,d3           *delay=(0-7)
330:         move.w  d3,4(a5)        *save delay
331:
332:         bra     loop_chk        *return to main loop
333:
334: atq_op:
335:         move.l  a1,lwk
336:         move.l  d1,dwk
337:
338:         moveq.l #0,d5
339:         move.b  3(a3),d5
340:         beq.s   non_op          *@Q=0なら関係無し
341:
342:         bsr     get_kc          *d1.w=kc
343:
344:         bsr     get_lng
345:
346:         tst.b   tie_flg
347:         bmi.s   non_op          *tieなら関係無し
348:
349:         sub.w   d5,number
350:         bmi.s   non_op
351:
352:         move.b  #'@',(a2)+
353:         move.b  #'L',(a2)+
354:
355:         move.w  number,d0
356:         bsr     num_wrt
357:         move.b  d1,d0           *d0=+,-
358:         lsr.w   #8,d1           *d1=kc
359:
360:         move.b  d1,(a2)+        *wrt_kc
361:         cmpi.b  #$20,d0
362:         beq.s   _r
363:         move.b  d0,(a2)+
364: _r:
365:         move.b  #'@',(a2)+
366:         move.b  #'L',(a2)+
367:         move.b  3(a3),d0        *write @Q length
368:         bsr     num_wrt
369:         move.b  #'R',(a2)+      *休符
370:
371:         bsr     okuru
372:         bra     loop_chk
373: non_op:
374:         movea.l lwk,a1
375:         move.l  dwk,d1
376:         bra     just_wrt
377:
378: modulation:                     *モジュレーション
379:         tst.b   5(a3)
380:         beq     atq_op          *lfo_swがoffなら...
381:
382:         moveq.l #0,d1
383:         moveq.l #0,d3
384:         moveq.l #0,d4
385:         moveq.l #0,d5
386:         moveq.l #0,d6
387:
388:         clr.b   oct_flg         *reset octave flag
389:
390:         bsr     get_kc
391:
392:         move.w  d1,d0
393:         bsr     where           *ｷｰから数値へ変換
394:         addq.b  #2,d4           *w1=w1+2
395:         move.l  d4,d6           *d6=w1
396:
397:         move.w  0(a5),d5        *d5=n1
398:         move.w  2(a5),d4        *d4=spd
399:
400:         bsr     get_lng
401:
402:         move.l  d5,d3           *d3=n1
403:         divu    d4,d3           *d3=n1/spd=a
404:
405:         move.w  (a4),d4         *d4=b=start kf
406:         cmpi.w  #-1,d4          *last kfが-1なら新たに
407:         beq.s   new_kf          *default kf をｾｯﾄ
408:         clr.w   dly_num
409:         bra.s   m1_m2
410: new_kf:
411:         moveq.l #0,d4
412:         move.b  (a3),d4
413: m1_m2:
414:         moveq.l #0,d0
415:         move.b  (a3),d0 *d0=default kf
416:
417:         move.w  d0,m1
418:         add.w   d5,m1           *m1=n1+kf
419:
420:         move.w  d0,m2
421:         sub.w   d5,m2           *m2=-n1+kf
422:
423:         bsr     okuru
424:
425:         move.w  dly_num,d0
426:         beq.s   start_mod
427:         move.b  #'@',(a2)+
428:         move.b  #'L',(a2)+
429:         bsr     num_wrt
430:
431:         bsr     wrt_kcd
432:         move.b  #'&',(a2)+      *tie
433: start_mod:
434:         move.b  #'@',(a2)+
435:         move.b  #'L',(a2)+
436:         move.b  #'1',(a2)+      *@L1
437:
438:         move.b  #'|',(a2)+
439:         move.b  #':',(a2)+      *|:     loop com
440:
441:         move.w  number,d5
442:         sub.w   dly_num,d5      *d5=length counter
443:         bmi     mod_err         *too short
444:         move.w  2(a5),d2
445:         lsl.w   #2,d2           *d2=spd*4 (波形の周期)
446:
447:         cmp.w   d2,d5
448:         bcs.s   mijikai
449: *周期=<音長
450:         andi.l  #$ffff,d5
451:         divu    d2,d5
452:         move.w  d5,d0
453:         bsr     num_wrt         *loop counter
454:
455:         move.l  d5,-(sp)
456:         move.w  d2,d5           *周期分,波形を生成
457:         subq.l  #1,d5           *dbra用
458:         bsr     mod_main
459:
460:         move.b  #':',(a2)+
461:         move.b  #'|',(a2)+      *loop end
462:
463:         move.l  (sp)+,d5
464:         swap    d5              *d5.w=あまり
465:         tst.w   d5
466:         beq.s   exit_mod        *丁度ぴったしの時...
467:
468:         subq.l  #1,d5           *dbra用
469:         bsr     mod_main
470:         bra.s   exit_mod
471: *周期>音長
472: mijikai:
473:         move.b  #'1',(a2)+      *loop=1
474:
475:         subq.l  #1,d5           *dbra用
476:         bsr     mod_main
477:
478:         move.b  #':',(a2)+
479:         move.b  #'|',(a2)+      *loop end
480:
481: exit_mod:
482:         tst.b   tie_flg
483:         bne.s   case_tie        *if tie then...
484: *case not tie
485:         move.b  #'y',(a2)+
486:         move.b  #'8',(a2)+
487:         move.b  #',',(a2)+
488:         moveq.l #0,d0
489:         move.b  ch_num,d0
490:         bsr     num_wrt
491:
492:         move.w  #-1,(a4)
493:         tst.b   q_flg
494:         bne     return
495:         move.b  #'@',(a2)+
496:         move.b  #'L',(a2)+
497:         move.b  3(a3),d0        *d0=@q
498:         bsr     num_wrt
499:         move.b  #'R',(a2)+
500:         bra     return2
501: case_tie:
502:         move.w  d4,(a4)         *save last kf
503:         bra     return          *帰還処理へ
504:
505: mod_main:
506:         moveq.l #0,d0
507:         move.b  6(a3),d0                *d0=sg
508: mod_lp:
509:         bsr     wrt_kcd         *音符を書く
510:
511:         move.w  d5,d7
512:         beq.s   do_calc_b       *ﾗｽﾄなら改行しない
513:         andi.b  #7,d7
514:         bne.s   do_calc_b
515:         bsr     okuru           *文字列が長くなったのでひとまず送る
516: do_calc_b:
517:         bsr     calc_b          *b=b+a*sg
518:
519:         cmp.w   m1,d4           *if b>m1 or b<m2 then...
520:         ble.s   chk_next
521:         bra.s   neg_sg
522: chk_next:
523:         cmp.w   m2,d4
524:         bge.s   next
525: neg_sg:
526:         neg.b   d0
527:         bsr     calc_b
528:         bsr     calc_b          *b=b+a*sg*2
529: next:
530:         dbra    d5,mod_lp
531:         move.b  d0,6(a3)        *save d0
532:         rts
533:
534: get_kc:
535:         * > d1= key code
536:         move.b  d0,d1           *d1=kc(capital)
537:         lsl.w   #8,d1
538:         move.b  #$20,d1         *space
539:         move.b  (a1),d2         調号?
540:         cmpi.b  #'#',d2
541:         bne.s   chk_nxt
542:         move.b  #'+',d2
543:         bra.s   install
544: chk_nxt:
545:         cmpi.b  #'+',d2
546:         beq.s   install
547:         cmpi.b  #'-',d2
548:         bne.s   exit_get_kc
549: install:
550:         move.b  d2,d1           *set +- to kc
```

▶９月号のLIVE in'90のラジオ体操，やりますねぇ。歌詞にいきなり背伸び123万4567回だもんなあ。体操やり終えて気づいたけどこれは123万4567回でなくて1，2，3，4，5，6，7，なんだよ。え，みんな知ってたの，ひどいなあ，教えてくれてもいいじゃないか。足の裏から血が出たよ。ビュッ！　　杉山　弘師（18）広島県

```
551:            addq.l  #1,a1               *next str
552: exit_get_kc:
553:            rts
554:
555: get_lng:
556:            bsr     num_chk
557:            beq.s   get_d_l             *case no number
558:            bsr     num_get             *d3=length
559:            cmpi.b  #192,d3
560:            bhi     l_err_in_macro      *too short
561:            tst.b   d3
562:            beq.s   _384                *0なら2倍全音符
563:            move.l  #192,d0
564:            divu    d3,d0               *d0=192/L
565:            move.w  d0,d3               *d0=d3=絶対音長
566:            bra.s   exit_gl
567: _384:
568:            move.w  #384,d3
569:            bra.s   exit_gl
570: get_d_l:
571:            move.b  1(a3),d3
572:            andi.w  #$00ff,d3
573:            move.w  d3,d0
574: exit_gl:
575:            bsr     futen               *符点ﾁｪｯｸ
576:            bsr     _@q
577:            move.w  d3,number
578:            lsr.w   #3,d3               *d3=d3/8
579:            mulu    4(a5),d3            *d3*dly
580:            move.w  d3,dly_num
581:            rts
582: _@q:
583:            move.b  #-1,q_flg
584:
585:            bsr     skip_spc
586:            cmpi.b  #'&',(a1)
587:            seq.b   tie_flg
588:            bne.s   _@q1
589:            addq.l  #1,a1               *skip '&'
590:            bra.s   ext_@q
591: _@q1:
592:            move.w  d3,dwk
593:            moveq.l #0,d0
594:            move.b  3(a3),d0
595:            beq.s   ext_@q              *case @q=0
596:            sub.w   d0,d3
597:            beq.s   non_@q
598:            smi.b   q_flg               *ｴﾗｰならq_flg=-1
599:            bmi.s   non_@q
600:            rts
601: non_@q:
602:            move.w  dwk,d3
603: ext_@q
604:            rts
605:
606: futen:
607:            * < d3=d0=length
608:            * < a1=str point
609:            * > d3=calculated length
610:            * > a1=next str
611:            * X d0
612: futen_lp:
613:            cmpi.b  #'.',(a1)
614:            bne.s   exit_ftn
615:            lsr.w   d0                  *d0=d0/2
616:            add.w   d0,d3
617:            addq.l  #1,a1
618:            bra.s   futen_lp
619: exit_ftn:
620:            rts
621:
622: wrt_kcd:                               *実際に音符を書く
623:            * < d4=bb
624:            * < d6=w1
625:            movem.l d0/d4/d6,-(sp)
626:            moveq.l #0,d2
627:
628:            tst.w   d4
629:            bmi.s   minus_lp            *d2=n,d4=bb
630: plus_lp:
631:            cmpi.w  #63,d4
632:            ble.s   wrtkcd1
633:            subi.w  #63,d4
634:            addq.b  #1,d2
635:            bra.s   plus_lp
636: minus_lp:
637:            tst.w   d4
638:            bpl.s   wrtkcd1
639:            addi.w  #63,d4
640:            subq.b  #1,d2
641:            bra.s   minus_lp
642: wrtkcd1:
643:            move.b  #'y',(a2)+
644:            moveq.l #48,d0
645:            add.b   ch_num,d0
646:            bsr     num_wrt
647:            move.b  #',',(a2)+
648:
649:            move.b  d4,d0               *d0=bb
650:            lsl.b   #2,d0               *kf=kf*4
651:            bsr     num_wrt             *write kf
652:
653:            add.b   d2,d6               *w1=w1+n
654:            andi.w  #$ff,d6
655:            move.b  oct_flg,d0
656:
657:            cmpi.b  #1,d6
658:            bhi.s   o_up?
659:            cmpi.b  #'>',d0
660:            beq.s   modosu?
661:            move.b  #'>',(a2)+
662:            move.b  #'>',oct_flg
663:            bra.s   yonbai
664: o_up?:
665:            cmpi.b  #14,d6
666:            bcs.s   modosu?
667:            cmpi.b  #'<',d0
668:            beq.s   modosu?
669:            move.b  #'<',(a2)+
670:            move.b  #'<',oct_flg
671:            bra.s   yonbai
672: modosu?:
673:            tst.b   d0
674:            beq.s   yonbai
675:            cmpi.b  #'>',d0
676:            bne.s   gyaku
677:            cmpi.b  #1,d6
678:            bls.s   yonbai
679:            move.b  #'<',(a2)+
680:            clr.b   oct_flg
681:            bra.s   yonbai
682: gyaku:
683:            cmpi.b  #14,d6
684:            bcc.s   yonbai
685:            move.b  #'>',(a2)+
686:            clr.b   oct_flg
687: yonbai:
688:            add.w   d6,d6               *d6=d6*2
689:
690:            lea     kc_tbl3,a0
691:
692:            move.b  0(a0,d6.w),(a2)+
693:            move.b  1(a0,d6.w),d0
694:            cmpi.b  #$20,d0
695:            beq.s   exit_wrtkcd         *spaceが出てきたら終わり
696:            move.b  d0,(a2)+
697:
698: exit_wrtkcd:
699:            movem.l (sp)+,d0/d4/d6
700:            rts
701:
702: calc_b:
703:            tst.b   d0                  *tst (sg)
704:            bpl.s   case_plus
705: *case_minus
706:            sub.w   d3,d4               *b=b-a
707:            bra.s   exit_calc_b
708: case_plus:
709:            add.w   d3,d4               *b=b+a
710: exit_calc_b:
711:            rts
712:
713: portament:
714:            clr.b   and_flg
715:            clr.b   oct_flg
716:            moveq.l #0,d1
717:
718:            move.b  (a1)+,d0            *d0=kc1
719:            bsr     moji_chk
720:            beq     err_in_macro        *error!
721:            move.b  d0,d1               *d1=kc1(capital)
722:            lsl.w   #8,d1
723:            move.b  #$20,d1             *space
724:            move.b  (a1)+,d2            *調号?
725:            cmpi.b  #'#',d2
726:            bne.s   chk_pm
727:            move.b  #'+',d2
728:            bra.s   torikomi
729: chk_pm:
730:            cmpi.b  #'+',d2
731:            beq.s   torikomi
732:            cmpi.b  #'-',d2
733:            bne.s   non_torikomi
734: torikomi:                              *第1ｷｰに調号を取り込む
735:            move.b  d2,d1               *d1=kc,cg
736:            move.b  (a1)+,d2
737: non_torikomi:
738:                                        *この時点でd2には"<,>"
739:            move.b  (a1)+,d0            *d0=kc2
740:            bsr     moji_chk
741:            beq     err_in_macro        *error!
742:            move.b  d0,d5               *d5=kc2
743:            lsl.w   #8,d5
744:            move.b  #$20,d5             *space
745:            move.b  (a1)+,d4            *調号?
746:            cmpi.b  #'#',d4
747:            bne.s   chk_pm2
748:            move.b  #'+',d4
749:            bra.s   torikomi2
750: chk_pm2:
751:            cmpi.b  #'+',d4
752:            beq.s   torikomi2
753:            cmpi.b  #'-',d4
754:            bne.s   non_torikomi2
755: torikomi2:                             *第2ｷｰに調号を取り込む
756:            move.b  d4,d5               *d5=kc,cg
757:            addq.l  #1,a1
758: non_torikomi2:
759:                                        *この時点でa1はlengthを指している
760:            bsr     num_chk
761:            beq.s   get_dflt_l
762:            bsr     num_get             *d3=length
763:            cmpi.l  #max,d3
764:            bhi     l_err_in_macro      *too long!
765:            cmpi.l  #1,d3
766:            bls     l_err_in_macro      *too short!
767:            bra.s   save_lng
768: get_dflt_l:
769:            move.b  1(a3),d3
770:            andi.w  #$00ff,d3
771: save_lng:
772:            move.w  d3,number           *save length
773:
774:            bsr     skip_spc
775:            cmpi.b  #'&',(a1)
776:            seq.b   tie_flg
777:            bne.s   not_tie
778:            addq.l  #1,a1
779: not_tie:                               *この時点でa1は次をﾎﾟｲﾝﾄ
780:            move.w  d1,d0
781:            bsr     where
782:            move.l  d4,d6               *d6=w1
783:            move.w  d5,d0
784:            bsr     where               *d4=w2
785:            cmpi.b  #'>',d2
786:            bne.s   _up?
787:            sub.b   #12,d4              *ｵｸﾀｰﾌﾞｽｲｯﾁ処理
788:            bra.s   _nxt
789: _up?:
790:            cmpi.b  #'<',d2
791:            bne.s   _nxt
792:            add.b   #12,d4              *ｵｸﾀｰﾌﾞｽｲｯﾁ処理
793: _nxt:
794:            move.b  (a3),d1 *d1=bend start
795:            sub.b   d6,d4
796:            tst.b   d4
797:            bmi.s   sg_minus
798:                                        *case_plus
799:            moveq.l #1,d2               *d2=sg=1
800:            bra.s   calc_range
801: sg_minus
802:            neg.b   d4                  *abs(d4)
803:            moveq.l #-1,d2              *d2=-1
804: calc_range:
805:            andi.w  #$00ff,d4
806:            mulu    #63,d4              *d4=d4*63
807:            add.w   d1,d4               *d4=bend end
808:            move.w  d4,d5
```

▶アルバイト青年，ついに挫折！　日中45℃を越える職場に重労働の熱処理工場は，あまくなかった。これでまたX68000君購入は，夢となったのであった。残念！

坊農　誠（19）福井県

```
809:         sub.w   d1,d5             *d5=d5-d1 (be-bs)
810:         bpl.s   decide_@L
811:         neg.b   d5                *d5=bend range
812: decide_@L:                        *経験的処理によって@Lを決定。
813:                                   *人工知能だぜ(大うそ)
814:         move.w  #1,L_work         *default is @L1
815:         cmpi.w  #exp,d3
816:         bls.s   prepare           *経験的処理は行わない
817: case_2?:
818:         move.w  d3,d0
819:         andi.w  #1,d0             *L mod 2
820:         bne.s   case_3?
821:         cmpi.w  #192,d5
822:         bcc.s   case_3?
823:         move.w  #2,L_work
824:         move.b  #-1,and_flg
825:         bra.s   prepare
826: case_3?:
827:         move.l  d3,d0
828:         andi.l  #$ffff,d0
829:         divu    #3,d0
830:         andi.l  #$ffff0000,d0
831:         bne.s   case_4?
832:         cmpi.w  #128,d5
833:         bcc.s   case_4?
834:         move.w  #3,L_work
835:         move.b  #-1,and_flg
836:         bra.s   prepare
837: case_4?:
838:         move.w  d3,d0
839:         andi.w  #3,d0             *L mod 4
840:         bne.s   prepare
841:         cmpi.w  #64,d5
842:         bcc.s   prepare
843:         move.w  #4,L_work
844:         move.b  #-1,and_flg
845: prepare:
846:         bsr     okuru
847:
848:         move.b  d1,d3             *d3=kf
849:         tst.b   d2                *check d2 (sg)
850:         bpl.s   prp1
851:         subq.b  #1,d6             *w1=w1-1
852:         add.b   #63,d3            *d3=63+kf
853: prp1:
854:         cmpi.b  #63,d3
855:         ble.s   prp2
856:         sub.b   #63,d3
857:         addq.b  #1,d6
858: prp2:
859:         tst.b   d6                *if w1<0 then...
860:         bpl.s   prp3
861:         moveq.l #11,d6            *w1=11
862:         move.b  #'>',(a2)+
863: prp3:
864:         cmpi.b  #11,d6            *if w1>11 then...
865:         ble.s   prp4
866:         moveq.l #0,d6             *w1=0
867: prp4:
868:         move.b  #'@',(a2)+
869:         move.b  #'L',(a2)+
870:         move.w  L_work,d0
871:         bsr     num_wrt           *write '@L'
872:
873:         move.w  number,d0
874:         subq.w  #1,d0             *d0=L-1
875:         move.w  d0,d7             *d7=L-1 (後で使う)
876:         move.w  d5,d1
877:         andi.l  #$ffff,d1         *d1=b
878:         divu    d0,d1             *d1=b/(L-1)=s
879:         move.l  d1,d4
880:         swap    d4                *d4=余り
881:         andi.l  #$ffff,d4
882:         lsl.l   #8,d4             *d4=余り*256
883:         divu    d0,d4             *d4=a
884:         move.l  d4,d0
885:         swap    d0
886:         tst.w   d0
887:         beq.s   calc_s
888:         addq.b  #1,d4             *d4=true a
889: calc_s:
890:         mulu    d2,d1             *s=s*sg
891:         move.w  L_work,d5
892:         moveq.l #0,d0
893:
894: port_lp:
895:         subq.b  #1,d5
896:         bne.s   non_wrt
897:         bsr     write_kc
898:         move.w  L_work,d5         *d5=lw
899: non_wrt:
900:         add.b   d4,d0             *n=n+a
901:         bcc.s   inc_y
902:                                   *case n>256
903:         add.b   d2,d3             *y=y+sg
904: inc_y:
905:         add.b   d1,d3             *y=y+s
906:
907:         move.w  d7,-(sp)
908:         beq.s   non_crlf          *ﾗｽﾄならは改行しない
909:         andi.b  #7,d7
910:         bne.s   non_crlf
911:         bsr     okuru             *文字列が長くなったのでひとまず送る
912: non_crlf:
913:         move.w  (sp)+,d7
914:         dbra    d7,port_lp
915:
916:         move.w  #-1,(a4)
917:
918:         tst.b   tie_flg
919:         bne.s   wr_@L             *if tie then...
920: *case not tie
921:         tst.b   and_flg
922:         beq.s   k_off_com
923:         subq.l  #1,a2             *delete last '&'
924:         bra.s   wr_@L
925: k_off_com:                        *@L1ならkey off comを送る
926:         move.b  #'y',(a2)+
927:         move.b  #'8',(a2)+
928:         move.b  #',',(a2)+
929:         moveq.l #0,d0
930:         move.b  ch_num,d0
931:         bsr     num_wrt
932: return:                           *帰還処理
933: wr_@L:
934:         move.b  #'@',(a2)+
935:         move.b  #'L',(a2)+
936:         moveq.l #0,d0
937:         move.b  1(a3),d0          *default @L
938:         bsr     num_wrt
939: return2:                          *ｵｸﾀｰﾌﾞのつじつま合わせ
940:         move.b  oct_flg,d0
941:         beq.s   rtn_end
942:         cmpi.b  #'>',d0
943:         beq.s   hantai
944:         move.b  #'>',(a2)+
945:         bra.s   rtn_end
946: hantai:
947:         move.b  #'<',(a2)+
948: rtn_end:
949:         bsr     okuru2
950:         bra     loop_chk
951:
952: okuru:
953:         andi.b  #%11111011,ccr
954:         bra.s   _okuru
955: okuru2:
956:         ori.b   #%00000100,ccr
957: _okuru:
958:         * < z   強制的にﾊﾞｯﾌｧへ転送
959:         *   nz  行の長さをﾁｪｯｸしてﾊﾞｯﾌｧへ転送
960:         movem.l d0-d7/a0-a7,-(sp)
961:
962:         lea     buffer,a1
963:
964:         beq.s   do_okuru
965:
966:         movea.l a2,a0
967:         suba.l  a1,a0
968:         cmpa.l  #line_max,a0
969:         bcs.s   exit_okuru2
970: do_okuru:
971:         clr.b   (a2)              *write string data end mark
972:
973:         suba.l  a1,a2
974:         move.l  a2,mj_lng
975:
976:         move.l  trk_num,d2
977:
978:         tst.b   out_flg
979:         beq.s   tsujou
980:         bsr     file_out
981:         bra.s   exit_okuru
982: tsujou:
983:         moveq.l #$06,d1
984:         IOCS    _OPMDRV           *ﾏｸﾛｺﾏﾝﾄﾞで生成したmmlには
985:                                   *誤りは無いものとする
986: exit_okuru:
987:         movem.l (sp)+,d0-d7/a0-a7
988:
989:         lea     buffer,a2
990:         rts
991: exit_okuru2:
992:         movem.l (sp)+,d0-d7/a0-a7
993:         rts
994:
995: write_kc:                         *ｷｰｺｰﾄﾞを実際に書く
996:         move.l  d0,-(sp)          *push   d0
997:
998:         cmpi.b  #63,d3
999:         ble.s   wrkc1
1000:        sub.b   #63,d3
1001:        addq.b  #1,d6
1002: wrkc1:
1003:        tst.b   d3
1004:        bpl.s   wrkc2
1005:        add.b   #63,d3
1006:        subq.b  #1,d6
1007: wrkc2:
1008:        tst.b   d6
1009:        bpl.s   wrkc3
1010:        moveq.l #11,d6
1011:        move.b  #'>',(a2)+
1012: wrkc3:
1013:        cmpi.b  #11,d6
1014:        ble.s   wrkc4
1015:        moveq.l #0,d6
1016:        move.b  #'<',(a2)+
1017: wrkc4:
1018:        move.b  #'y',(a2)+
1019:        moveq.l #48,d0
1020:        add.b   ch_num,d0
1021:        bsr     num_wrt
1022:        move.b  #',',(a2)+
1023:
1024:        cmpi.b  #63,d3
1025:        bne.s   case_not_63
1026: *case y=63
1027:        move.b  #'0',(a2)+        *y48+n,0
1028:
1029:        lea     kc_tbl2,a0
1030:        move.w  d6,d0             *d0=w1
1031:        addq.b  #1,d0
1032:        add.b   d0,d0             *d0=(w1+1)*2
1033:        adda.l  d0,a0
1034:
1035:        move.b  0(a0),(a2)+
1036:        move.b  1(a0),d0
1037:        cmpi.b  #$20,d0
1038:        beq.s   exit_wr_kc
1039:        move.b  d0,(a2)+
1040:        bra.s   exit_wr_kc
1041:
1042: case_not_63:
1043:        move.b  d3,d0
1044:        lsl.b   #2,d0             *y=y*4 (0-252)
1045:        bsr     num_wrt
1046:
1047:        lea     kc_tbl,a0
1048:        move.w  d6,d0             *d0=w1
1049:        add.b   d0,d0             *d0=w1*2
1050:        adda.l  d0,a0
1051:
1052:        move.b  0(a0),(a2)+
1053:        move.b  1(a0),d0
1054:        cmpi.b  #$20,d0
1055:        beq.s   exit_wr_kc
1056:        move.b  d0,(a2)+
1057:
1058: exit_wr_kc:
1059:        tst.b   and_flg
1060:        beq.s   exit_wr_kc1
1061:        move.b  #'&',(a2)+        *@L1以外なら'&'が必要
1062: exit_wr_kc1:
1063:        move.l  (sp)+,d0          *pop    d0
1064:        rts
1065:
1066: where:          *search kc
```

▶狭いんです，部屋が！ 4.5畳1間にX68000，ステレオ，29型TV，ビデオテープの山，タンスなど……。いーかげん寝るとこがなくなってきた。どーしよ？

安藤 弘明 (21) 福島県

```
1067:         * < d0 search kc
1068:         * > d4 kc number(0-11)
1069:         * X
1070:
1071:         move.l  a0,lwk
1072:         lea     kc_tbl,a0
1073:         moveq.l #0,d4
1074:         bsr     wh_lp
1075:         beq.s   exit_wh
1076:         lea     kc_tbl1,a0
1077:         moveq.l #0,d4
1078:         bsr     wh_lp
1079:         bne     kc_err
1080: exit_wh:
1081:         movea.l lwk,a0
1082:         rts
1083:
1084: wh_lp:
1085:         cmp.w   (a0)+,d0
1086:         beq.s   exit_wh_lp      *z
1087:         addq.b  #1,d4
1088:         tst.b   (a0)
1089:         bne.s   wh_lp
1090:         andi.b  #%11111011,ccr  *nz
1091: exit_wh_lp:
1092:         rts
1093:
1094: moji_chk:       *d0がキーコードかどうかチェック
1095:         * < d0
1096:         * > z   not kc
1097:         * > nz  kc
1098:         * > d0 大文字で出てくる
1099:
1100:         andi.b  #$df,d0
1101:         cmpi.b  #'A',d0
1102:         bcs.s   exit_chk_mj
1103:         cmpi.b  #'G',d0
1104:         bhi.s   exit_chk_mj
1105:         tst.b   d0              *make not zero
1106:         rts
1107: exit_chk_mj:
1108:         ori.b   #$ff,ccr        *強制的にｾﾞﾛﾌﾗｸﾞｾｯﾄ
1109:         rts
1110:
1111: num_chk:        *(a1)が数字かどうかチェック
1112:         * < (a1)
1113:         * > z   not number
1114:         * > nz  number
1115:         * X
1116:
1117:         cmpi.b  #'0',(a1)
1118:         bcs.s   exit_chk_num
1119:         cmpi.b  #'9',(a1)
1120:         bhi.s   exit_chk_num
1121:         tst.b   (a1)            *make not zero
1122:         rts
1123: exit_chk_num:
1124:         ori.b   #$ff,ccr        *強制的にｾﾞﾛﾌﾗｸﾞｾｯﾄ
1125:         rts
1126:
1127: l_wk_set1:
1128:         move.b  d1,(a2)+        *まず'L'を書く...
1129:         move.l  a1,-(sp)
1130:         bsr     num_get         *d3=length number
1131:         move.l  #192,d0
1132:         divu    d3,d0           *d0.w=@L length
1133:
1134:         move.b  d0,1(a3)        *ﾁｬﾝﾈﾙﾜｰｸに書く
1135: exit_lwk:
1136:         move.l  (sp)+,a1
1137:         bra     loop_chk
1138:
1139: skip_spc:                       *無駄なスペースカット
1140:         cmpi.b  #$20,(a1)
1141:         bne.s   exit_skp_spc
1142:         addq.l  #1,a1
1143:         bra.s   skip_spc
1144: exit_skp_spc:
1145:         rts
1146:
1147: v_wk_set1:
1148:         move.b  d1,(a2)+        *まず'V'を書く...
1149:         move.l  a1,-(sp)
1150:         bsr     num_get         *d3=volume value
1151:         lea     v_tbl,a1
1152:         move.b  (a1,d3.w),d0    *@V value
1153:
1154:         move.b  d0,2(a3)        *ﾁｬﾝﾈﾙﾜｰｸに書く
1155:         move.l  (sp)+,a1
1156:         bra     loop_chk
1157:
1158: wk_set2:
1159:         move.b  d1,(a2)+        *write '@'
1160:         move.b  (a1)+,(a2)+     *write 'l' , 'L' , 'v' or 'V'
1161:         movem.l d7/a1,-(sp)
1162:         bsr     num_get         *d3=@length number
1163:         move.b  d3,(a3,d6.w)    *ﾁｬﾝﾈﾙﾜｰｸに書く
1164:         movem.l (sp)+,d7/a1
1165:         bra     loop_chk
1166:
1167: detune:
1168:         move.b  #'y',(a2)+      *write y_com
1169:         bsr     num_chk
1170:         bne.s   num_frm_bf
1171:         moveq.l #0,d3
1172:         move.b  (a3),d3
1173:         bra.s   do_w
1174: num_frm_bf:
1175:         bsr     num_get         *d3=number(0-63)
1176:
1177:         cmpi.b  #63,d3
1178:         bhi.w   err_in_macro    *if d3>63 then...
1179:
1180:         move.b  d3,(a3) *ｷｰﾌﾗｸｼｮﾝの値(0-63)をﾁｬﾝﾈﾙﾜｰｸへ
1181: do_w:
1182:         lsl.b   #2,d3           *d3=d3*4  (0-252)
1183:
1184:         moveq.l #0,d0           *reset d0
1185:         move.b  ch_num,d0       *d0=ch_num
1186:         add.b   #$30,d0         *d0=(48-55)
1187:         bsr     num_wrt
1188:
1189:         move.b  #',',(a2)+      *write ','
1190:
1191:         move.l  d3,d0
1192:         bsr     num_wrt
1193:
1194:         bra.w   loop_chk
1195:
1196: num_wrt:                        *bufferに数字を書く
1197:         * < d0=書きたい値(0-65535)
1198:         * < a2=書き込みたいアドレス
1199:         * > a2=next addr
1200:         * X
1201:         movem.l d0/d4,-(sp)
1202:         clr.b   (wrt_flg)
1203:         move.l  #10000,d4
1204:         andi.l  #$ffff,d0       *d0=0-65535
1205: nmw_lp:
1206:         divu    d4,d0   *d0=d0/d4
1207:         tst.b   (wrt_flg)
1208:         bne.s   kanarazu_w      *初めてじゃないのなら必ず書く
1209:         tst.b   d0              *初めてのケース
1210:         beq.s   no_wrt
1211:         move.b  #$ff,(wrt_flg)
1212: kanarazu_w:
1213:         add.b   #$30,d0
1214:         move.b  d0,(a2)+        *1文字ライト
1215: no_wrt:
1216:         swap    d0      *d0=余り
1217:         andi.l  #$ffff,d0
1218:         divu    #10,d4  *d4=d4/10
1219:         bne.s   nmw_lp
1220:         tst.b   (wrt_flg)
1221:         beq.s   wrt_zero
1222:         movem.l (sp)+,d0/d4
1223:         rts
1224: wrt_zero:
1225:         move.b  #'0',(a2)+
1226:         movem.l (sp)+,d0/d4
1227:         rts
1228:
1229: num_get:
1230:         * < a1=数字列の存在するアドレス
1231:         * > d3=値
1232:         * > a1=next addr
1233:         * X
1234:         move.l  d0,-(sp)
1235:         moveq.l #0,d0
1236:         moveq.l #0,d3
1237: num_lp:
1238:         move.b  (a1)+,d0
1239:         sub.b   #$30,d0
1240:         bmi.s   num_exit
1241:         cmpi.b  #$9,d0
1242:         bhi.s   num_exit
1243:
1244:         mulu    #10,d3  *d3=d3*10
1245:         add.w   d0,d3   *d3=d3+d0
1246:         bra.s   num_lp
1247: num_exit:
1248:         move.l  (sp)+,d0
1249:         subq.l  #1,a1
1250:         rts
1251:
1252: m_switch:
1253:         tst.b   15(sp)
1254:         sne.b   out_flg
1255:         clr.l   d0
1256:         bsr     junbi
1257:         clr.l   d0
1258:         rts
1259:
1260: file_out:
1261:         tst.b   opn_flg
1262:         bne.s   file_out1       *既にファイルをオープンしてある
1263:         move.w  #32,-(sp)
1264:         pea     filename
1265:         DOS     _CREATE
1266:         addq.l  #6,sp
1267:         move.w  d0,fh           *save file handle
1268:         bmi     file_out_err
1269:         move.b  #-1,opn_flg
1270: file_out1:
1271:         lea     sub_buf,a2
1272:         move.b  #'(',(a2)+
1273:         move.b  #'t',(a2)+
1274:         move.w  d2,d0
1275:         bsr     num_wrt
1276:         move.b  #')',(a2)+
1277:         suba.l  #sub_buf,a2
1278:         pea     (a2)
1279:         pea     sub_buf
1280:         move.w  fh,-(sp)
1281:         DOS     _WRITE
1282:         lea     10(sp),sp
1283:         tst.l   d0
1284:         bmi     file_out_err
1285:
1286:         move.l  mj_lng,-(sp)
1287:         pea     buffer
1288:         move.w  fh,-(sp)
1289:         DOS     _WRITE
1290:         lea     10(sp),sp
1291:         tst.l   d0
1292:         bmi     file_out_err
1293:
1294:         moveq.l #13,d0
1295:         bsr     fputc
1296:         moveq.l #10,d0
1297: fputc:
1298:         move.w  fh,-(sp)
1299:         move.w  d0,-(sp)
1300:         DOS     _FPUTC
1301:         addq.l  #4,sp
1302:         rts
1303:
1304: file_out_err:
1305:         DOS     _ALLCLOSE
1306:         moveq.l #55,d0
1307:         bra.s   err_msg_sel
1308: too_long_err:
1309:         moveq.l #54,d0
1310:         bra.s   err_msg_sel
1311: rv_err:
1312:         moveq.l #53,d0
1313:         bra.s   err_msg_sel
1314: mod_err:
1315:         moveq.l #52,d0
1316:         bra.s   err_msg_sel
1317: kc_err:
1318:         move.b  d0,er_ky+1
1319:         lsr.w   #8,d0
1320:         move.b  d0,er_ky
1321:
1322:         moveq.l #51,d0
1323:         bra.s   err_msg_sel
1324:
```


▶“プロミストランド”っていいですね。あの重苦しい歌声，あの厳しい歌詞。えっ，浜田省吾じゃないんですか？
陣山　達夫（20）大阪府

```
1325: l_err_in_macro:
1326:         moveq.l #50,d0          *length error
1327:         bra.s   err_msg_sel
1328:
1329: err_in_macro:
1330:         moveq.l #49,d0          *macro error
1331:         bra.s   err_msg_sel
1332: junbi:
1333:         lea.l   modorichi(pc),a0
1334:         move.l  d0,$0006(a0)
1335:         rts
1336: err?:
1337:         tst.l   err_flg
1338:         bpl     ext_err?
1339:         moveq.l #$09,d0         *OPMDRV未登録エラー
1340:         bra     err_msg_sel
1341: ext_err?:
1342:         rts
1343:
1344: err_msg_sel:
1345:         movea.l svsp,sp
1346:         move.l  d0,-(a7)
1347:         subq.l  #1,d0
1348:         asl.l   #2,d0
1349:         movea.l err_cd_tbl(pc,d0.l),a1
1350:         moveq.l #$ff,d0
1351:         bsr     junbi
1352:         move.l  (a7)+,d0
1353:         rts
1354:
1355: err_cd_tbl:
1356:         .dc.l   tempo_err
1357:         .dc.l   trk_num_err
1358:         .dc.l   cmd_err
1359:         .dc.l   size_err
1360:         .dc.l   mem_err
1361:         .dc.l   channel_err
1362:         .dc.l   dim_err
1363:         .dc.l   inst_num_err
1364:         .dc.l   drv_err
1365:         .dc.l   0               *10
1366:         .dc.l   0
1367:         .dc.l   0
1368:         .dc.l   0
1369:         .dc.l   0
1370:         .dc.l   0
1371:         .dc.l   0
1372:         .dc.l   0
1373:         .dc.l   0
1374:         .dc.l   mml_err
1375:         .dc.l   kakko_err       *20
1376:         .dc.l   no_kakko
1377:         .dc.l   repeat_err
1378:         .dc.l   no_rpt_num
1379:         .dc.l   rpt_num_err
1380:         .dc.l   no_oct_num
1381:         .dc.l   octv_err
1382:         .dc.l   length_err
1383:         .dc.l   out_of_mem
1384:         .dc.l   inst_err
1385:         .dc.l   no_@w_num       *30
1386:         .dc.l   _@w_err
1387:         .dc.l   no_tmp_num
1388:         .dc.l   tmp_num_err
1389:         .dc.l   no_l_num
1390:         .dc.l   no_v_num
1391:         .dc.l   v_num_err
1392:         .dc.l   no_kc_num
1393:         .dc.l   kc_num_er
1394:         .dc.l   inst_num_err
1395:         .dc.l   rnp_kakko_nashi *40
1396:         .dc.l   too_many_keys
1397:         .dc.l   no_key
1398:         .dc.l   q_err
1399:         .dc.l   rnp_mml_err
1400:         .dc.l   y_com_err
1401:         .dc.l   _@L_err
1402:         .dc.l   pan_err
1403:         .dc.l   tie_err
1404:         .dc.l   macro_err
1405:         .dc.l   l_err           *50
1406:         .dc.l   kc_err_msg
1407:         .dc.l   mod_err_msg
1408:         .dc.l   rv_err_msg
1409:         .dc.l   too_long_msg
1410:         .dc.l   file_out_err_msg
1411: wrt_flg:
1412:         .ds.b   1
1413: tie_flg:
1414:         .ds.b   1
1415: ch_num:
1416:         .ds.b   1
1417: oct_flg:
1418:         .ds.b   1
1419:
1420: v_tbl:
1421:         .dc.b   85,87,90,93,95,98,101,103,106,109,111,114,117,119,122,125
1422: and_flg:
1423:         .ds.b   1
1424: q_flg:
1425:         .ds.b   1
1426: out_flg:
1427:         .ds.b   1
1428: opn_flg:
1429:         .ds.b   1
1430: filename:
1431:         .dc.b   'ZMUSIC.TMP',0
1432:         .even
1433: lwk:
1434:         .ds.l   1
1435: dwk:
1436:         .ds.l   1
1437: kc_tbl:
1438:         .dc.b   'C C+D D+E F F+G G+A A+B ',0,0  *必ず大文字!!
1439:                                                 *Bの後のspace!!
1440: kc_tbl1:
1441:         .dc.b   'C D-D E-E F G-G A-A B-B ',0,0  *必ず大文字!!
1442:                                                 *Bの後のspace!!
1443: kc_tbl2:
1444:         .dc.b   'C C+D D+E F F+G G+A A+B <C',0,0
1445: kc_tbl3:
1446:         .dc.b   'B-B C C+D D+E F F+G G+A A+B C C+'
1447: number:
1448:         .ds.l   1
1449: dly_num:
1450:         .ds.l   1
1451: L_work:
1452:         .ds.w   1
1453: trk_num:
1454:         .ds.l   1
1455: m1:
1456:         .dc.w   0
1457: m2:
1458:         .dc.w   0
1459: svsp:
1460:         .ds.l   1
1461: fh:
1462:         .ds.w   1
1463: mj_lng:
1464:         .ds.l   1
1465: modorichi:
1466:         .dc.b   $00,$00,$00,$00,$00,$00
1467:         .ds.b   16              *return buffer
1468: err_flg:
1469:         .dc.l   $00000000
1470: tempo_err:
1471:         .dc.b   'テンポの指定が無効です',$00
1472: trk_num_err:
1473:         .dc.b   'トラック番号が無効です',$00
1474: cmd_err:
1475:         .dc.b   'ｍｕｓｉｃ　ｃｍｄ　ｅｒｒｏｒ',$00
1476: size_err:
1477:         .dc.b   'サイズの指定が無効です',$00
1478: mem_err:
1479:         .dc.b   'メモリー確保できません',$00
1480: channel_err:
1481:         .dc.b   'チャンネル番号が無効です',$00
1482: dim_err:
1483:         .dc.b   '配列の指定に誤りがあります',$00
1484: drv_err:
1485:         .dc.b   '音楽用関数は無効です',$00
1486: mml_err:
1487:         .dc.b   'ＭＭＬの文法に誤りがあります',$00
1488: kakko_err:
1489:         .dc.b   '［］の中の指定に誤りがあります',$00
1490: no_kakko:
1491:         .dc.b   '］がありません',$00
1492: repeat_err:
1493:         .dc.b   '繰り返しの値が無効です',$00
1494: no_rpt_num:
1495:         .dc.b   '繰り返し番号の指定がありません',$00
1496: rpt_num_err:
1497:         .dc.b   '繰り返し番号の指定が無効です',$00
1498: no_oct_num:
1499:         .dc.b   'オクターブの番号がありません',$00
1500: octv_err:
1501:         .dc.b   'オクターブの指定が無効です',$00
1502: length_err:
1503:         .dc.b   '長さの値が無効です',$00
1504: out_of_mem:
1505:         .dc.b   'トラック容量が足りません',$00
1506: inst_err:
1507:         .dc.b   '音色番号の指定がありません',$00
1508: no_@w_num:
1509:         .dc.b   '＠Ｗの値がありません',$00
1510: _@w_err:
1511:         .dc.b   '＠Ｗの値が無効です',$00
1512: no_tmp_num:
1513:         .dc.b   'テンポの値がありません',$00
1514: tmp_num_err:
1515:         .dc.b   'テンポの値が無効です',$00
1516: no_l_num:
1517:         .dc.b   '長さの値がありません',$00
1518: no_v_num:
1519:         .dc.b   '音量の値がありません',$00
1520: v_num_err:
1521:         .dc.b   '音量の値が無効です',$00
1522: no_kc_num:
1523:         .dc.b   'キーコードの値がありません',$00
1524: kc_num_er:
1525:         .dc.b   'キーコードの値が無効です',$00
1526: inst_num_err:
1527:         .dc.b   '音色番号が無効です',$00
1528: rnp_kakko_nashi:
1529:         .dc.b   '｝がありません',$00
1530: too_many_keys:
1531:         .dc.b   '｛｝の中に音符が多すぎます',$00
1532: no_key:
1533:         .dc.b   '｛｝の中に音符がありません',$00
1534: q_err:
1535:         .dc.b   'Ｑの指定に誤りがあります',$00
1536: rnp_mml_err:
1537:         .dc.b   '｛｝の中の文法に誤りがあります',$00
1538: y_com_err:
1539:         .dc.b   'Ｙの指定に誤りがあります',$00
1540: _@L_err:
1541:         .dc.b   '＠Ｌの指定に誤りがあります',$00
1542: pan_err:
1543:         .dc.b   'Ｐの指定に誤りがあります',$00
1544: tie_err:
1545:         .dc.b   'タイの指定に誤りがあります',$00
1546: macro_err:
1547:         .dc.b   'ﾏｸﾛｺﾏﾝﾄﾞの使用法に誤りがあります',$00
1548: l_err:
1549:         .dc.b   'ﾏｸﾛｺﾏﾝﾄﾞに対しての音長指定に誤りがあります',$00
1550: kc_err_msg:
1551:         .dc.b   'ﾏｸﾛｺﾏﾝﾄﾞ中に誤ったｷｰｺｰﾄﾞ（'
1552: er_ky:
1553:         .dc.b   '  '
1554:         .dc.b   '）が用いられています',$00
1555: mod_err_msg:
1556:         .dc.b   'ﾓｼﾞｭﾚｰｼｮﾝ･ﾏｸﾛｺﾏﾝﾄﾞ中に誤りがあります',$00
1557: rv_err_msg:
1558:         .dc.b   '相対ﾎﾞﾘｭｰﾑ･ﾏｸﾛｺﾏﾝﾄﾞ中に誤りがあります',$00
1559: too_long_msg:
1560:         .dc.b   '繰り返し回数が大きいか、繰り返しの範囲が広すぎます',$00
1561: file_out_err_msg:
1562:         .dc.b   'ファイル作成に失敗しました',$00
1563:         .even
1564: wk_sz:  equ     8
1565: wk_sz2: equ     3               *ﾜｰｸｻｲｽﾞが2の何乗か
1566: ch_work:        *format( kf, @L, @V, @Q, _~, lfosw, sgn, dmy)
1567:         .ds.b   8*wk_sz
1568:
1569: parm:   equ     8               *number of parameters
1570: parm2:  equ     3               *parmが2の何乗か
1571: lfo_wk:
1572:         .ds.b   parm*8
1573: last_kf:
1574:         .ds.w   8
1575: sub_buf:
1576:         .ds.b   8
1577: buffer:
1578:         .ds.b   1024
1579:         .end
```

▶グラナダをやっていると，吐き気がしませんか？　とくに4面あたりで。だれかループレバー作って！
橘　正彦 (17) 福島県

リスト7　ZMUSIC.FNCを利用した「スーパー忍」

日本音楽著作権協会(出)許諾第9071289-001号

```
10 /*************************************************************/
20 /*                                                           */
30 /*           T H E   S U P E R   S H I N O B I               */
40 /*                                                           */
50 /*                      THE SHINOBI                          */
60 /*                                                           */
70 /*                (C) SEGA / YUZO.KOSHIRO                    */
80 /*                                                           */
90 /*                ARRANGED BY Z.NISHIKAWA                    */
100 /*                                                           */
110 /*************************************************************/
120 m_init()
130 dim char v(4,10)
140 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN          琴
150 v={ 56,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
160 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
170     26,   8,   5,   7,   2,  30,   3,   3,   3,   0,   0,
180     29,   5,   4,   4,   1,  30,   3,   4,   3,   0,   0,
190     28,   4,   2,   6,   3,  30,   3,   1,   3,   0,   0,
200     31,  10,   3,   5,   1,   0,   3,   1,   3,   0,   0}
210 m_vset(70,v)
220 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN       E.BASS
230 v={ 33,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
240 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
250     31,  20,   0,   7,   5,  30,   0,   7,   0,   0,   0,
260     31,  10,   0,   7,   5,  20,   0,   0,   0,   0,   0,
270     31,  14,   0,   7,  10,  23,   0,   0,   0,   0,   0,
280     31,  31,   0,   7,   0,   0,   0,   0,   0,   0,   0}
290 m_vset(71,v)
300 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN      TRUMPET
310 v={ 58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
320 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
330     13,  14,   0,   3,   1,  23,   2,   1,   0,   0,   0,
340     13,  14,   0,  10,  15,  40,   2,   7,   0,   0,   0,
350     14,  14,   0,   3,   1,  38,   2,   1,   0,   0,   0,
360     19,   3,   0,  10,   0,   0,   1,   1,   0,   0,   1}
370 m_vset(72,v)
380 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN     HH CLOSE
390 v={ 61,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
400 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
410     26,  10,   5,  10,  10,   0,   2,  14,   0,   3,   0,
420     31,  20,  11,  15,  11, 127,   1,   3,   7,   1,   0,
430     28,  24,  10,  15,  10,   2,   0,   4,   7,   1,   0,
440     19,  21,  12,  15,  11,   0,   0,   2,   7,   1,   0}
450 m_vset(73,v)
460 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN      EFFECTS
470 v={  0,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
480 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
490      6,   5,   0,  15,   3,  16,   0,   0,   6,   0,   0,
500      6,   0,   0,  15,   2,  25,   0,   2,   3,   0,   0,
510      8,   0,   0,  15,   2,  26,   0,   2,   3,   0,   0,
520     16,   4,   0,  15,   3,   4,   0,   0,   3,   0,   0}
530 m_vset(74,v)
540 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN      STRINGS
550 v={ 44,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
560 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
570     31,   0,   0,   0,   0,  33,   0,   8,   7,   0,   0,
580     16,  15,   0,   6,   3,   8,   0,   8,   7,   0,   1,
590     31,   0,   0,   0,   0,  24,   0,   4,   3,   0,   0,
600     31,  15,   0,   6,   3,   4,   0,   4,   3,   0,   1}
610 m_vset(75,v)
620 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN     SYNTHE 1
630 v={ 56,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
640 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
650     16,  14,   0,   3,   3,  26,   0,   3,   6,   0,   0,
660     19,   0,   0,   5,   0,  25,   0,   1,   3,   0,   0,
670     26,   0,   0,   3,   0,  26,   0,   1,   3,   0,   0,
680     27,   0,   0,  11,   0,   0,   0,   1,   3,   0,   0}
690 m_vset(76,v)
700 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN      E.PIANO
710 v={ 58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
720 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
730     28,   4,   0,   5,   0,  37,   1,   1,   7,   0,   0,
740     22,   9,   0,   2,   0,  47,   1,  12,   0,   0,   0,
750     29,   4,   0,   6,   0,  37,   1,   3,   3,   0,   0,
760     31,  10,   0,   5,  10,   0,   1,   1,   0,   0,   0}
770 m_vset(77,v)
780 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN       MELODY
790 v={ 58,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
800 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
810     20,   0,   0,   6,   0,  37,   0,   0,   0,   0,   0,
820     20,   0,   0,   6,   0,  47,   0,   4,   0,   0,   0,
830     20,   0,   0,   6,   0,  42,   0,   0,   0,   0,   0,
840     14,  10,   0,   8,   0,   0,   0,   1,   0,   0,   1}
850 m_vset(78,v)
860 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN      COWBELL
870 v={ 59,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
880 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
890     31,  21,  19,   6,   2,   0,   0,  15,   6,   0,   0,
900     31,  21,  12,   6,   2,  35,   0,   8,   6,   1,   0,
910     31,  21,  13,   6,   3,  32,   0,   7,   0,   0,   0,
920     31,  19,  16,   9,   2,   0,   0,   2,   0,   0,   0}
930 m_vset(79,v)
940 /*  AF   OM   WF   SY   SP  PMD  AMD  PMS  AMS  PAN           鼓
950 v={ 59,  15,   0,   0,   0,   0,   0,   0,   0,   3,   0,
960 /*  AR   DR   SR   RR   SL   OL   KS   ML  DT1  DT2  AME
970     31,   0,   0,   5,   0,  12,   0,  14,   0,   0,   0,
980     31,  10,   0,   5,  15,  60,   0,   0,   0,   0,   0,
990     27,  27,   0,  10,  15,  37,   0,  10,   0,   0,   0,
1000    28,  14,   0,   7,  15,   0,   1,   0,   0,   0,   0}
1010 m_vset(80,v)
1020 str a[256],b[256],c[256],d[256]
1030 str a1[256],b1[256],c1[256],d1[256]
1040 str a2[256],b2[256],c2[256],d2[256]
1050 str x[256],y[256],z[256],b3[256],c3[256]
1060 int i
1070 /*
1080 str bd="y2,23",sd="y2,15",sd0="y2,16",sd1="y2,14"
1090 /*str bd="y2,23",sd="y2,21",sd0="y2,22",sd1="y2,20"  /*こっちで
もいいよ
1100 str p1="y3,1",p2="y3,2",p3="y3,3"
1110 /*
1120 for i=1 to 8:m_alloc(i,8000):m_assign(i,i):next
1130 /*
1140 m_tempo(112)
1150 /*
1160 for i=1 to 8:m_trk2(i,"[do]"):next
1170 /*
1180 key 2,"m_play()"+chr$(13):key 12,"m_stop()"+chr$(13)
1190 /*
1200 /*         M E L O D Y
1210 /*
1220 a="L16|: g<dddddddd+d+d+d+d+d+ d+ffffffffd+d+d+d+d+|1d+d+>:|
1230 b="L8 =1 ^10,13,1 g a1<d2..c> a1& a&v10a&_3a&_a&_ar4 v12<d&(d,e)
24&e2.&(e,e-)24L16rdea2&a8.g8ede2.. =0 r@70 v12p1d>a+ag8.a+agd4.
1240 c="=1^10,13,4 L16g1<g1 d2.&d8.c+ c8.>a+8.<f2&f8>
1250 c=c+c
1260 d="=0 |: @72 o5 v12 q6 g8g8r8ggr2g8g8ggr8 r |1 @70 v10 o4 q8 f8.
ffr8:| r4..
1270 m_trk2(1,"@70 v12 q8 o4 k0"+a)
1280 m_trk2(1,"@78 v12    o5       "+b)
1290 m_trk2(1,"@75 v14    o2 k0 p1"+c)
1300 m_trk2(1,d)
1310 /*
1320 /*         S U B
1330 /*
1340 b="L8 =1 ^10,14,1 rg a1<d2..c> a1& a&v8a&_3a&_a&_ar4 v10<d&(d,e)
24&e2.&(e,e-)24L16rdea2&a8.g8ede2.. =0 r@70 v10p1d>a+ag8.a+agd4
1350 d="=0 |: @76 o2 v13 q4 g8g8r8ggr2g8g8ggr8 r4 |1 @70 v11 o5 q8 ff
r8:|r4
1360 m_trk2(2,"@76 v9  q8 o2 k8"+a)
1370 m_trk2(2,"@78 v10   o5    "+b)
1380 m_trk2(2,"@75 v14    o2 k10 p2"+c)
1390 m_trk2(2,d)
1400 /*
1410 /*         B A S S
1420 /*
1430 a="L16|:  @71v14g4v11ggv12gv14grgr4. g4v11ggv14ggrgr8.@77p1<d8.>
:|
1440 b="|:4 @71v14g4v11ggv12gv14grgr4. g4v11ggv14ggrgr8.@77p1<d8.>:|
1450 c="|: ggggr<g8>grgrg8gaa+ <d+d+d+d+r<d+8>d+rd+rd+8d>a+8 g8<g>gg<
g>gg<fgcd>gaa+g< d+8<d+>d+d+<d+>d+d+<c+d+>g+a+c+d+>d+f :|
1460 d="|: g8g8r8ggr4rfdfg8g8ggr8 |1 r2 :| v14gdfg<cdfg
1470 m_trk2(3,"         q8 o2 k0 "+a)
1480 m_trk2(3,b)
1490 m_trk2(3,"@71 v14    o2       "+c)
1500 m_trk2(3,"    v15    o2       "+d)
1510 /*
1520 /*         C H O R D   &   B A C K I N G
1530 /*
1540 a="L16|:d32&>a32&g&g2..r1:|
1550 b="L16|:4 @77 o3 q6 p2 v13 d4 v12 p3 <a+r8.araara8.r4 @72 o5 p1
v13 a+r8ararar4:|
1560 c="L16|:rcccrdddrd+d+d+rfff rgggc4rd+d+d+>a+4< d8d4dd2&d d+8d+4d
+d+2&d+:|
1570 d="|: @77 o5 v14 c8c8r8ccr2c8c8ccr8 r8.|1 @70 v11 o5 cccr8:| rr4
1580 m_trk2(4,"@72 v10 q8 o4 k0 "+a)
1590 m_trk2(4,"              k0 "+b)
1600 m_trk2(4,"@76 v9     o3       "+c)
1610 m_trk2(4,d)
1620 /*
1630 a="L2 v8a&~3a&~a&~a&~a&~a&~a&~a
1640 b="L16|:4 @77 o3 q6 p2 v13 g4 v12 p3 <fr8.ereere8.r4 @72 o5 p1 v
13 fr8ererer4:|
1650 c="L16|:rdddreeerfffrggg raaad4rfffc4 g8g4gg2&g f8f4ff2&f:|
1660 d="|: @77 o5 v14 d8d8r8ddr2d8d8ddr8 r8 |1 @70 v11 o4 a+8a+a+r8:|
r4.
1670 m_trk2(5,"@75      q8 o2 k0 "+a)
1680 m_trk2(5,b)
1690 m_trk2(5,"@77 v12    o4       "+c)
1700 m_trk2(5,d)
1710 /*
1720 a="L2 v8d&~3d&~d&~d&~d&~d&~d&~d
1730 b="L16|:4 @77 o2 q6 p2 v13 g4 v12 p3<<dr8.crccrc8.r4 @72 o5 p1 v
13 dr8crcrcr4:|
1740 c="L16|:rgggraaara+a+a+<rccc rddd>g4ra+a+a+f4 a8a4aa2&a a+8a+4a+
a+2&a+:|
1750 d="|: @77 o5 v14 k0 g8g8r8ggr2g8g8ggr8 r |1 @70 v10 o4 y53,20 f8
.ffr8:| r4..
1760 m_trk2(6,"@75      q8 o2 k0"+a)
1770 m_trk2(6,b)
1780 m_trk2(6,"@77 v12    o4       "+c)
1790 m_trk2(6,d)
1800 /*
1810 /*         E F F E C T S
1820 /*
1830 a="L1 |:v13g&v14g:|
1840 b="L16|:4 brbr <ere>b rbrb b8b8    b8br <eer>b rbbr <eer8> :|
1850 c="|: g1<g1 d2.&d8.c+ c8.>a+2.&a+:|"
1860 d="r1 r2L64<c&d&d+&e&f&g&a&b&<c&d&e&f&g&a&b&<cr4 r1r1
1870 m_trk2(7,"@74      q8 o3 k0 "+a)
1880 m_trk2(7,"@79 v15    o4       "+b)
1890 m_trk2(7,"@72 v8  q8 o4 k10"+c)
1900 m_trk2(7,"    v7     o2 k0 "+d)
1910 /*
1920 /*         D R U M S
1930 /*
1940 a="L16|:"+bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"cr"+bd+"r"
+sd+"c8"+bd+"r8"+bd+"c8"+bd+"r8"+sd+"c8|1"+bd+"r"+bd+"rc&"+bd+"c"+bd+"
r"+bd+"r"+sd+"c8"+bd+"r8:|
1950 a=a+bd+"r"+bd+"rc&"+sd0+"c"+sd0+"r"+sd+"r"+sd+"c&"+sd1+"c"+sd1+"
r"+sd1+"r"
1960 b1="|:3"+bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"cr"+bd+"r"+
sd+"c8"+bd+"r8"+bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"c"+bd+"r"+
bd+"r"+sd+"c8"+bd+"r8:|
1970 b2=bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"cr"+bd+"r"+sd+"c8
"+bd+"r8"+bd+"c8"+sd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+sd0+"c"+sd+"r"+sd1+"
r"+bd+"c&"+sd0+"c"+sd+"r"+sd1+"r"
1980 c1=bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"cc&"+bd+"c"+sd+"r
4" +bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"c"+bd+"c&"+bd+"c"+sd+"
r8"+bd+"r8"
1990 c2=bd+"cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"c@80o6p2v13a"+bd+"av14a"+b
d+"a"+sd+"av15aa@v127a" +bd+"@73o5v14cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"cc
"+bd+"r"+bd+"c"+bd+"c"+sd+"r8"+bd+"cc"
2000 c3=bd+"cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"c@80o6p2v13a"+bd+"av14a"+b
d+"a"+sd+"av15aa@v127a" +bd+"@73o5v14cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"cc
"+bd+"r"+bd+"c"+bd+"c"+sd+"r8"+bd+"c"+bd+"c"
2010 d1=bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"cc&"+bd+"c"+sd+"r
8"+bd+"r8" +bd+"c8"+bd+"r8"+sd+"c8"+bd+"r"+bd+"rc&"+bd+"c"+bd+"c&"+bd+
"c"+sd+"r8"+bd+"r8"
2020 d2=bd+"cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"cc"+bd+"rc"+bd+"c"+sd+"r8"
+bd+"cc" +bd+"cr"+bd+"cc"+sd+"r8"+bd+"c"+bd+"c" +bd+"c"+bd+"r"+bd+"c"+
bd+"c"+sd+"r"+sd+"r"+sd+"c"+sd+"c"
2030 m_trk2(8,"@73 o5 q8 v14 k0 "+p3+a)
2040 m_trk2(8,b1):m_trk2(8,b2)
2050 m_trk2(8,"|:"+c1):m_trk2(8,"|1"+c2+":|"):m_trk2(8,c3)
2060 m_trk2(8,d1):m_trk2(8,d2)
2070 for i=1 to 8:m_trk2(i,"[loop]"):next
2080 m_play()
```

付録　簡易OPMファイルジェネレータ

OMUSIC.FNC

Nishikawa Zenji 西川 善司

いままで作ったX-BASICのミュージックプログラムは「OPMファイル」の形式にしておくと，システム上でいつでも曲の演奏ができてたいへん便利です。短いプログラムで変換できますから，ぜひ入力して使ってみてください。

既存のBASICミュージックプログラムをOPMファイルに変換するプログラムを作成してみました。その名も（べつに凄んで言うまでもないけど）簡易OPMファイルジェネレータ「OMUSIC. FNC」。もうひとつ，機械語は嫌いだ！　という人向けにBASIC版も用意しました（こちらは多少使い勝手が悪い）。

入力と起動

リスト1のダンプリストをマシン語入力ツール（6月号付録ディスクのmac. xなど）から入力するか，またはリスト2のソースプログラムをエディタから入力しアセンブルしてください。以下の作業は基本的にZMUSIC. FNC（97ページ）の場合と同じ手順ですので初心者の方はそちらを参照してください。ここでは簡潔に示します。

また，ZMUSIC. FNCと同様DOSCALL. MACをインクルードする必要がありますのでご注意ください。

アセンブル，リンクを終了したら生成されたOMUSIC. XをOMUSIC. FNCにリネームし，自分のシステムディスクのBASICのディレクトリへコピーしてください。

次にBASICのコンフィギュレーションファイルを変更します。エディタでBASIC. CNFを読み込んでください。

画面から，

　FUNC＝MUSIC

の1行を削除し，いちばん下の行に，

　FUNC＝OMUSIC

を付け足してください。これで準備完了です。あとはいつもどおり，

　A＞BASIC

でOMUSIC. FNCを組み込んだBASICが起動します。

OMUSIC. FNCの使い方

使い方はいたって簡単です。OPMファイルへ変換したいBASICのミュージックプログラムをLOADしRUNするだけです。RUNするとカレントドライブのカレントディレクトリにTEMP. OPMというファイル名のOPMファイルを作成します。ドライブの準備ができていなかったりすると当然のことながらエラーが発生しますからご注意。また，OMUSIC.FNCとMUSIC. FNCは共存できません。MUSIC. FNCが先に登録されている場合はOPMファイルは作成されず，通常どおり音楽が演奏されるだけです。

さて，あとはシステムへ帰還し，できたTEMP. OPMを，お好みのファイルネーム（～. OPM）に変更するなりしてください。演奏法はすでにご存じですね，

　COPY ファイルネーム OPM

です。繰り返しいいますがOPMDRV. Xがデバイスドライバとして登録されていることが前提です。「デバイスドライバってなんじゃらほい？」なんていっている「こまったちゃん」は速やかに「Human68kユーザーズマニュアル」の「システムの構築」の項を開きましょうね。

BASIC版の入力方法と使用法

さて，次にBASIC版の説明をします。こちらは，BASICを普通に起動しリスト3を入力してください。入力を終えたら，

　SAVE@"MAKER. BAS"

でセーブしてください。SAVE@とはBASICプログラムを行番号なしでセーブする命令です。

では，前述のOMUSIC. FNCの起動法を参考にして「MUSIC. FNCを組み込まないBASIC」のシステムを起動してください（BASICコンフィギュレーションファイルからFUNC＝MUSICの1行を削除する）。MUSIC. FNCが未登録になっているか確認するには，ダイレクトに，

　m_init(　)

などとすれば，

　関数は定義されていません

というエラーメッセージが返ってくるはず

COMMAND. X上での演奏制御

ゲームミュージックなど，ミュージックプログラムの中にはMMLの［DO］～［LOOP］を使って無限ループで演奏するものがあります。いくら素晴らしい音楽でも，聴き飽きてしまったり，集中して思考を巡らせたいときなど，演奏を止めたいと思うことがあります。こんなときはどうしたらよいのでしょう。なんとかOPMDRV. Xへ演奏制御のコマンドを送ることが出来ないでしょうか。

皆さんはシステム予約ファイル「CON」というのをご存じでしょうか。これはCONSOLE（コンソール）の略で，キーボードからの入力またはスクリーンへの出力を指定するときに使用します。この予約ファイル名を駆使してOPMDRV. Xへコマンドを送ることができるのです。やり方はOPMファイルを演奏する要領で，

　A＞COPY CON OPM

です。これを実行するとBASICのINPUT命令のような入力待ち状態となりますので，前の方で説明した「OPMファイルフォーマット」のコマンドを入力してリターンキーを押してください。たとえば演奏を止めたいのなら，

　A＞COPY CON OPM
　(s)

です。

［BREAK］キーでこの状態を終了します。もちろん，演奏再開の「(c)」や初めから演奏させる「(p)」なども送ることができますよ。

私はいちいちCOPY命令を打ち込むのは面倒臭いので以下のようにしています。

　(s)

のように演奏制御命令のみを書いたOPMファイルを作成，このOPMファイルを「STOP. OPM」とし，今度は，

　ECHO OFF
　ECHO　演奏を中断します
　COPY STOP. OPM　OPM

のようなバッチファイルを作って，これを「E. BAT」とします。これで，演奏を止めたいときは，

　A＞E (RETURN)

の1文字でOK，ほとんど新しいコマンドが増えた気分です。同様に「演奏再開」や「演奏開始」も作っておくといいでしょう。

です（エラーなしで「OK」と出て実行できてしまった人は初めからやり直すように）。

では，好みのBASICミュージックプログラムをロードしてください。

ところでBASIC版は音楽用外部関数をfunc～endfuncによって作ったユーザー関数で代行し，これに出力されたパラメータをOPMファイルフォーマットに加工してファイルとして出力するものです。func～endfunc命令で作ったユーザー関数は配列変数のパラメータを引数として持つことができません。よって，配列変数を引数とするm_vset命令はどうしても再現できないのです。

そこで，BASIC版は，音色設定に使用する配列変数をグローバル変数「v」に統一することによって，むりやりm_vset命令をエミュレートします。そんなわけでOPMファイル変換対象のミュージックプログラムの音色設定用配列変数を「v（小文字）」に統一修正してください。

具体的には，たとえば，

```
dim char bass(4,10)={…
```

という変数で音色を設定しているならば，これを，

```
dim char v(4,10)={…
```

に，また，

```
bass={……
```

は，

```
v={……
```

に変更してほしいわけです。

次に，

```
search"m_vset"
```

で，プログラム中に使用されているmvset命令をリストアップし，パラメータを音色番号のみに（配列型の引数は削除）してください。たとえば，

```
m_vset(70, voice)
```

は配列変数であるvoiceをカットし，

```
m_vset(70)
```

というようにしてください。なお，ミュージックプログラムのなかには音色設定を行わず，プリセットの音色のみを使用しているものもあります。こういったものについては以上の作業を行う必要はありません。

今度は，プログラム先頭にo_openの1行を加えてください。多くの場合は，

```
1 o_open( )
```

でいいはずです。このo_openは以降のファイル出力に備え，TEMP.OPMというファイルをオープン（新規作成）する関数です。

ふぅー。これでやっと下準備が終わりました（マシン語版OMUSIC.FNCの場合は今までの作業をする必要はありません）。

では，ミュージックプログラムよりも後ろにMAKER.BASを，

```
LOAD@"MAKER.BAS",行番号
```

のようにしてロードします（必ず，MAKER.BASはミュージックプログラムよりも大きい行番号でロードしてください）。

無事ロードされたことを確認し，ディスクドライブの準備をしたらRUNしてください。カレントドライブのカレントディレクトリにTEMP.OPMというOPMファイルを作成します。

使用上の注意

演奏制御にm_stat命令を使用し，whileやgotoで無限ループを構成してあるようなミュージックプログラムをコンバートするとファイルサイズが「∞」になってしまう（すなわち永遠にファイル出力を続けてしまう）ので注意してください。たとえば，こんなやつです。

```
while 1
    while m_stat( )
    endwhile
    m_play( )
endwhile
```

こういったプログラムは，while文などを削除し，m_play( )のみにしてください。

「OPMファイルジェネレータ」は以下の命令のみをコンバートします。

```
m_alloc
m_assign
m_init
m_tempo
m_trk
m_vset
m_play( )
```

これ以外の命令，たとえばm_cont, m_stopはコンバートしません。使用していた場合はエラーが発生します。まあ，ほとんどのミュージックプログラムは上記の7つの命令で作られているはずですからなんら問題もないはずです（m_atoiなんて使った人見たことないよ，あたしゃ……）。

最後にもう一点だけ注意。ファイル操作をするようなミュージックプログラムのコンバートは避けてください，コンバートに失敗する可能性があります（そんなミュージックプログラム，あるわけないか）。

とりあえず，超短いミュージックプログラムを用意しました。リスト4がそれです。このプログラムを「BASIC版OPMファイルジェネレータ」でコンバートするために下準備処理を施したのがリスト5，OPMファイルにコンバートしたものがリスト6です。もちろん機械語版であるOMUSIC.FNCを使用してリスト4を直接コンバートしても，リスト6と同じものが作成されます。

ZMUSIC.FNCを使用したミュージックプログラムのコンバート

今月発表となったZMUSIC.FNCを使用したミュージックプログラムをOPMファイルにコンバートする手順を簡単に説明します。

まず，ZMUSIC.FNCとOMUSIC.FNCの2つの外部関数を組み込んだBASICを立ち上げてください。

次に，

```
m_switch(1)
```

を実行してください。このコマンドはZMUSIC.FNCに入っている関数で，m_trk2（同じくZMUSIC.FNCに内蔵の）命令に出力されたMMLをファイル出力するかどうかのスイッチ命令です。スイッチ＝1でファイル書き出しモード，＝0で通常動作モードです。

コンバートしたいミュージックプログラムをロードしたらディスクドライブの確認をしてRUNしてください。「ZMUSIC.TMP」と「TEMP.OPM」の2つのファイルが作成されるはずです。ZMUSIC.TMPはm_trk2によって出力されたMMLですのでこれをエディタなどでTEMP.OPMと結合して再セーブします。具体的には，

```
A>ED TEMP.OPM
```

でまず初めにOMUSIC.FNCで出力されたものをロードしてください。m_playすなわち「(p)」を画面から探し出しこの上にカーソルを合わせます。［ESC］+［Y］のファイル読み込みコマンドでZMUISC.TMPをロードしてください。うまくファイルが結合されたことを確認したら［ESC］+［E］でファイルをセーブ，エディタを終了してください。

```
A>COPY TEMP.OPM OPM
```

これで曲が鳴り出すはずです。

BASIC版のコンバータを使用してのコンバートも以上の手順と同様に行ってください。

OPMAやOPMDを使用したミュージックプログラムは

1989年4月号に発表された「OPMA」や本誌6月号の付録ディスクに収録された「OPMD」はFM音源とAD PCM（サンプリング音）を同期するドライバで，これらを駆使したミュージックプログラムがOh!X紙上に数多く発表されてきました。もちろんこれらのミュージックプログラムもOPMDRV.Xを使用して演奏されるわけですから今月のOMUSIC.FNCなどでOPMファイルへ問題なくコンバートが可能です。

OPMDを使用しているのであればMIDI用のミュージックプログラムであってもコンバートできます。

リスト1　ダンプ

(1514バイトでセーブ)

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 00 00 00 00 05 4E  : 53
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 00 5C 00 00 00 00  : 5C
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  00 00 00 F8 00 00 00 DC  : D4
0048  00 00 00 F8 00 00 00 F8  : F0
0050  00 00 00 F8 00 00 00 F8  : F0
0058  00 00 00 F8 00 00 00 F8  : F0
0060  00 00 00 40 00 00 00 74  : B4
0068  00 00 00 C0 00 00 00 00  : C0
0070  00 00 00 00 00 00 00 00  : 00
0078  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  48 55 00 3C 00 00 05 86  46FA

0080  6D 5F 61 73 73 69 67 6E  : 51
0088  00 6D 5F 61 6C 6C 6F 63  : D7
0090  00 6D 5F 76 73 65 74 00  : 8E
0098  6D 5F 74 72 6B 00 6D 5F  : E9
00A0  70 6C 61 79 00 6D 5F 69  : EB
00A8  6E 69 74 00 6D 5F 74 65  : F0
00B0  6D 70 6F 00 00 00 00 90  : DC
00B8  00 00 00 96 00 00 00 9C  : 32
00C0  00 00 00 A2 00 00 00 A8  : 4A
00C8  00 00 00 BA 00 00 00 BC  : 76
00D0  00 04 00 84 80 01 00 04  : 0D
00D8  00 02 80 01 00 04 00 54  : DB
00E0  80 01 00 04 00 08 80 01  : 0E
00E8  00 84 00 84 00 84 00 84  : 10
00F0  00 84 00 84 00 84 00 84  : 10
00F8  80 01 80 01 00 04 80 01  : 87
-----------------------------------
SUM:  25 ED D7 B9 AA 1F 8A F0  2AC4

0100  00 00 01 02 00 00 01 32  : 36
0108  00 00 01 62 00 00 01 CA  : 2E
0110  00 00 01 F2 00 00 02 0A  : FF
0118  00 00 01 AA 3F 3C 00 20  : 46
0120  48 79 00 00 03 42 FF 3C  : 41
0128  5C 8F 33 C0 00 00 03 00  : E1
0130  5B F9 00 00 03 4C 4E 75  : 66
0138  42 39 00 00 03 4C FF 1F  : E8
0140  4E 75 45 F9 00 00 03 4D  : 51
0148  20 2F 00 0C 22 2F 00 16  : C2
0150  14 FC 00 28 14 FC 00 61  : A9
0158  61 00 01 3E 14 FC 00 2C  : DC
0160  30 01 61 00 01 34 14 FC  : D7
0168  00 29 61 00 00 F6 60 00  : E0
0170  01 80 45 F9 00 00 03 4D  : 0F
0178  20 2F 00 0C 22 2F 00 16  : C2
-----------------------------------
SUM:  75 B3 84 30 B5 96 CD 45  BD6B

0180  14 FC 00 28 14 FC 00 6D  : B5
0188  61 00 01 0E 14 FC 00 2C  : AC
0190  30 01 61 00 01 04 14 FC  : A7
0198  00 29 61 00 00 C6 60 00  : B0
01A0  01 50 45 F9 00 00 03 4D  : DF
01A8  20 2F 00 0C 20 6F 00 16  : 00
01B0  D1 FC 00 00 00 10 14 FC  : ED
01B8  00 28 14 FC 00 76 61 00  : 0F
01C0  00 D8 14 FC 00 2C 14 FC  : 24
01C8  00 30 14 FC 00 2C 74 36  : 16
01D0  10 18 61 00 00 C4 14 FC  : 5D
01D8  00 2C 51 CA FF F4 53 8A  : 17
01E0  14 FC 00 29 61 7C 60 00  : 76
01E8  01 08 45 F9 00 00 03 4D  : 97
01F0  20 2F 00 0C 14 FC 00 28  : 93
01F8  14 FC 00 6F 61 00 00 9A  : 7A
-----------------------------------
SUM:  F0 44 3B 96 1E 3F 3E BB  3E6B

0200  14 FC 00 29 61 5C 60 00  : 56
0208  00 E8 45 F9 00 00 03 4D  : 76
0210  22 6F 00 16 20 2F 00 0C  : 02
0218  14 FC 00 28 14 FC 00 74  : BC
0220  61 76 14 FC 00 29 14 D9  : FD
0228  66 FC 53 8A 61 34 60 00  : 34
0230  00 C0 45 F9 00 00 03 4D  : 4E
0238  14 FC 00 28 14 FC 00 70  : B8
0240  14 FC 00 29 61 1C 60 00  : 16
0248  00 A8 45 F9 00 00 03 4D  : 36
0250  14 FC 00 28 14 FC 00 69  : B1
0258  14 FC 00 29 61 04 60 00  : FE
0260  00 90 4A 39 00 00 03 4C  : 62
0268  6B 00 00 98 14 FC 00 0D  : 20
0270  14 FC 00 0A 48 52 95 FC  : 45
0278  00 00 03 4D 48 52 48 79  : AB
-----------------------------------
SUM:  E0 A5 83 A2 84 9C 7D E7  FEC6

0280  00 00 03 4D 3F 39 00 00  : C8
0288  03 00 FF 40 4F EF 00 0A  : 8A
0290  24 5F 4A 80 6B 72 4E 75  : ED
0298  48 E7 88 00 42 39 00 00  : 32
02A0  03 4B 28 3C 00 00 27 10  : E9
02A8  02 80 00 00 FF FF 80 C4  : C4
02B0  4A 39 00 00 03 4B 66 0C  : 43
02B8  4A 00 67 0E 13 FC 00 FF  : CD
02C0  00 00 03 4B D0 3C 00 30  : 8A
02C8  14 C0 48 40 02 80 00 00  : DE
02D0  FF FF 88 FC 00 0A 66 D6  : C8
02D8  4A 39 00 00 03 4B 67 06  : 3E
02E0  4C DF 00 11 4E 75 14 FC  : 0F
02E8  00 30 4C DF 00 11 4E 75  : 2F
02F0  70 00 61 04 70 00 4E 75  : 08
02F8  41 FA 00 30 21 40 00 06  : D2
-----------------------------------
SUM:  62 4B E3 02 04 F0 D8 56  3F50

0300  4E 75 58 8F 70 01 60 04  : 7F
0308  58 8F 70 02 48 E7 80 00  : 08
0310  53 80 E5 80 22 7B 08 0C  : E9
0318  70 FF 61 DC 4C DF 00 01  : D8
0320  4E 75 00 00 03 02 00 00  : C8
0328  03 23 00 00 00 00 00 00  : 26
0330  00 00 00 00 00 00 00 00  : 00
0338  00 00 00 00 00 00 00 00  : 00
0340  00 00 83 74 83 40 83 43  : 80
0348  83 8B 82 CC 83 49 81 5B  : 04
0350  83 76 83 93 82 C9 8E B8  : A0
0358  94 73 82 B5 82 DC 82 B5  : D3
0360  82 BD 00 83 66 81 5B 83  : 87
0368  5E 82 CC 8F 91 82 AB 8D  : 86
0370  9E 82 DD 82 C9 8E B8 94  : 22
0378  73 82 B5 82 DC 82 B5 82  : C1
-----------------------------------
SUM:  45 D2 76 8B CF 85 6F 42  6A88

0380  BD 00 54 45 4D 50 2E 4F  : 70
0388  50 4D 00 00 00 00 00 00  : 9D
0390  00 00 00 00 00 00 00 00  : 00
0398  00 00 00 00 00 00 00 00  : 00
03A0  00 00 00 00 00 00 00 00  : 00
03A8  00 00 00 00 00 00 00 00  : 00
03B0  00 00 00 00 00 00 00 00  : 00
03B8  00 00 00 00 00 00 00 00  : 00
03C0  00 00 00 00 00 00 00 00  : 00
03C8  00 00 00 00 00 00 00 00  : 00
03D0  00 00 00 00 00 00 00 00  : 00
03D8  00 00 00 00 00 00 00 00  : 00
03E0  00 00 00 00 00 00 00 00  : 00
03E8  00 00 00 00 00 00 00 00  : 00
03F0  00 00 00 00 00 00 00 00  : 00
03F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  0D 4D 54 45 4D 50 2E 4F  2F87
```

057F$_H$まで0で埋める

```
0580  00 00 00 00 00 00 00 00  : 00
0588  00 00 00 00 00 00 00 00  : 00
0590  00 04 00 04 00 04 00 04  : 10
0598  00 04 00 04 00 04 00 04  : 10
05A0  00 04 00 04 00 4C 00 04  : 58
05A8  00 04 00 04 00 04 00 04  : 10
05B0  00 04 00 34 00 04 00 04  : 40
05B8  00 04 00 04 00 04 00 04  : 10
05C0  00 0A 00 0A 00 06 00 08  : 22
05C8  00 0A 00 30 00 30 00 48  : B2
05D0  00 20 00 28 00 18 00 18  : 78
05D8  00 14 00 08 00 06 00 18  : 3A
05E0  00 14 00 0E 00 1A 00 48  : 84
05E8  00 04 00 00 00 00 00 00  : 04
05F0  00 00 00 00 00 00 00 00  : 00
05F8  00 00 00 00 00 00 00 00  : 00
-----------------------------------
SUM:  00 78 00 C0 00 CE 00 E0  7B5F
```

リスト2　ソース

```
  1: **********************************************
  2: *                                            *
  3: *        簡易ＯＰＭ　ＦＩＬＥジェネレーター        *
  4: *                                            *
  5: *               BY   Z.NISHIKAWA             *
  6: *                                            *
  7: **********************************************
  8:
  9:         .include        doscall.mac
 10:         .text
 11:
 12: info_tbl:
 13:         .dc.l   init
 14:         .dc.l   run
 15:         .dc.l   end
 16:         .dc.l   system
 17:         .dc.l   break
 18:         .dc.l   ctrl_d
 19:         .dc.l   yobi
 20:         .dc.l   yobi
 21:         .dc.l   token_tbl
 22:         .dc.l   param
 23:         .dc.l   exec_adr
 24:         .ds.b   20
 25:
 26: token_tbl:
 27:         .dc.b   'm_assign',$00
 28:         .dc.b   'm_alloc',$00
 29:         .dc.b   'm_vset',$00
 30:         .dc.b   'm_trk',$00
 31:         .dc.b   'm_play',$00
 32:         .dc.b   'm_init',$00
 33:         .dc.b   'm_tempo',$00
 34: param:
 35:         .dc.l   masgn_p
 36:         .dc.l   mallc_p
 37:         .dc.l   mvset
 38:         .dc.l   mtrk
 39:         .dc.l   mplay
 40:         .dc.l   minit
 41:         .dc.l   tempo
 42: masgn_p:
 43:         .dc.w   $0004,$0084,$8001
 44: mallc_p:
 45:         .dc.w   $0004,$0002,$8001
 46: mvset:
 47:         .dc.w   $0004,$0054,$8001
 48: mtrk:
 49:         .dc.w   $0004,$0008,$8001
 50: mplay:
 51:         .dc.w   $0084,$0084,$0084,$0084
 52:         .dc.w   $0084,$0084,$0084,$0084
 53:         .dc.w   $8001
 54: minit:
 55:         .dc.w   $8001
 56: tempo:
 57:         .dc.w   $0004,$8001
 58: exec_adr:
 59:         .dc.l   m_assign
 60:         .dc.l   m_alloc
 61:         .dc.l   m_vset
 62:         .dc.l   m_trk
 63:         .dc.l   m_play
 64:         .dc.l   m_init
 65:         .dc.l   m_tempo
 66:
 67: run:
 68:         move.w  #32,-(sp)
 69:         pea     filename                *file open
 70:         DOS     _CREATE
 71:         addq.l  #6,sp
 72:         move.w  d0,fh
 73:         smi.b   err_flg
 74:         rts
 75: init:
 76: ctrl_d:
 77: system:
 78: break:
 79: yobi:
 80: end:
 81:         clr.b   err_flg
 82:         DOS     _ALLCLOSE
 83:         rts
 84:
 85: m_assign:
 86:         lea     buffer,a2
 87:         move.l  $000c(a7),d0            *ﾁｬﾝﾈﾙ
 88:         move.l  $0016(a7),d1            *get trk number
 89:         move.b  #'(',(a2)+
 90:         move.b  #'a',(a2)+
 91:         bsr     num_wrt                 *ch
 92:         move.b  #',',(a2)+
 93:         move.w  d1,d0
 94:         bsr     num_wrt                 *trk
 95:         move.b  #')',(a2)+
 96:         bsr     fwrts
 97:         bra     bye_bye
 98:
 99: m_alloc:
100:         lea     buffer,a2
101:         move.l  $000c(a7),d0
102:         move.l  $0016(a7),d1
103:         move.b  #'(',(a2)+
104:         move.b  #'m',(a2)+
105:         bsr     num_wrt                 *tr
106:         move.b  #',',(a2)+
107:         move.w  d1,d0
108:         bsr     num_wrt                 *size
109:         move.b  #')',(a2)+
110:         bsr     fwrts
111:         bra     bye_bye
112: m_vset:
113:         lea     buffer,a2
114:         move.l  $000c(a7),d0            *d0=v number
115:         movea.l $0016(a7),a0
116:         adda.l  #$00000010,a0           *a0=ﾃﾞｰﾀが実際にあるアドレス
117:         move.b  #'(',(a2)+
118:         move.b  #'v',(a2)+
```

```
119:         bsr      num_wrt                     *v number
120:         move.b   #',',(a2)+
121:         move.b   #'0',(a2)+                  *dummy
122:         move.b   #',',(a2)+
123:         moveq.l  #55-1,d2
124: vst_lp:
125:         move.b   (a0)+,d0                    *get one data
126:         bsr      num_wrt
127:         move.b   #',',(a2)+
128:         dbra     d2,vst_lp
129:         subq.l   #1,a2                       *delete last ','
130:         move.b   #')',(a2)+
131:         bsr      fwrts
132:         bra      bye_bye
133: m_tempo:
134:         lea      buffer,a2
135:         move.l   $000c(a7),d0
136:         move.b   #'(',(a2)+
137:         move.b   #'o',(a2)+
138:         bsr      num_wrt                     *tempo
139:         move.b   #')',(a2)+
140:         bsr      fwrts
141:         bra      bye_bye
142: m_trk:
143:         lea      buffer,a2
144:         movea.l  $0016(a7),a1                *mml data
145:         move.l   $000c(a7),d0                *track num
146:         move.b   #'(',(a2)+
147:         move.b   #'t',(a2)+
148:         bsr      num_wrt
149:         move.b   #')',(a2)+
150: mtr_lp:
151:         move.b   (a1)+,(a2)+
152:         bne.s    mtr_lp
153:         subq.l   #1,a2
154:         bsr      fwrts
155:         bra      bye_bye
156: m_play:
157:         lea      buffer,a2
158:         move.b   #'(',(a2)+
159:         move.b   #'p',(a2)+
160:         move.b   #')',(a2)+
161:         bsr      fwrts
162:         bra      bye_bye
163: m_init:
164:         lea      buffer,a2
165:         move.b   #'(',(a2)+
166:         move.b   #'i',(a2)+
167:         move.b   #')',(a2)+
168:         bsr      fwrts
169:         bra      bye_bye
170: fwrts:
171:         tst.b    err_flg
172:         bmi      fopen_err
173:         move.b   #13,(a2)+
174:         move.b   #10,(a2)+                   *cr lf
175:         pea      (a2)                        *save a2
176:         suba.l   #buffer,a2
177:         pea      (a2)
178:         pea      buffer
179:         move.w   fh,-(sp)
180:         DOS      _WRITE
181:         lea      10(sp),sp
182:         movea.l  (sp)+,a2
183:         tst.l    d0
184:         bmi      fwrite_err
185:         rts
186:
187: num_wrt:                              *bufferに数字を書く
188:         * < d0=書きたい値(0-65535)
189:         * < a2=書き込みたいアドレス
190:         * > a2=next addr
191:         * X
192:         movem.l d0/d4,-(sp)
193:         clr.b    (wrt_flg)
194:         move.l   #10000,d4
195:         andi.l   #$ffff,d0        *d0=0-65535
196: nmw_lp:
197:         divu     d4,d0   *d0=d0/d4
198:         tst.b    (wrt_flg)
199:         bne.s    kanarazu_w       *初めてじゃないのなら必ず書く
200:         tst.b    d0               *初めてのケース
201:         beq.s    no_wrt
202:         move.b   #$ff,(wrt_flg)
203: kanarazu_w:
204:         add.b    #$30,d0
205:         move.b   d0,(a2)+         *1文字ライト
206: no_wrt:
207:         swap     d0      *d0=余り
208:         andi.l   #$ffff,d0
209:         divu     #10,d4  *d4=d4/10
210:         bne.s    nmw_lp
211:         tst.b    (wrt_flg)
212:         beq.s    wrt_zero
213:         movem.l  (sp)+,d0/d4
214:         rts
215: wrt_zero:
216:         move.b   #'0',(a2)+
217:         movem.l  (sp)+,d0/d4
218:         rts
219: bye_bye:
220:         moveq.l  #0,d0
221:         bsr      junbi
222:         moveq.l  #0,d0
223:         rts
224: junbi:
225:         lea.l    modorichi(pc),a0
226:         move.l   d0,$0006(a0)
227:         rts
228:
229: fopen_err:
230:         addq.l   #4,sp
231:         moveq.l  #1,d0            *err_num
232:         bra.s    err_msg_sel
233: fwrite_err:
234:         addq.l   #4,sp
235:         moveq.l  #2,d0            *err_num
236: err_msg_sel
237:         movem.l  d0,-(a7)
238:         subq.l   #1,d0
239:         asl.l    #2,d0
240:         movea.l  err_cd_tbl(pc,d0.l),a1
241:         moveq.l  #$ff,d0
242:         bsr      junbi
243:         movem.l  (a7)+,d0
244:         rts
245: err_cd_tbl:
246:         .dc.l    fopen_err_msg
247:         .dc.l    fwrite_err_msg
248:
249: modorichi:
250:         .dc.b    $00,$00,$00,$00,$00,$00
251: rtn_buffer:
252:         .ds.b    16               *return buffer
253: fh:
254:         .ds.w    1                *file handle
255: fopen_err_msg:
256:         .dc.b    'ファイルのオープンに失敗しました',$00
257: fwrite_err_msg:
258:         .dc.b    'データの書き込みに失敗しました',$00
259: filename:
260:         .dc.b    'TEMP.OPM',$00
261: wrt_flg:
262:         .ds.b    1
263: err_flg:
264:         .ds.b    1
265: buffer:
266:         .ds.b    512
```

リスト3 BASIC版

```
 1: end
 2: func o_open()
 3: str fn[256]="TEMP.OPM"
 4: fopen(fn,"c")
 5: endfunc
 6: func m_init()
 7: fwrites("(i)"+chr$(13)+chr$(10),0)
 8: endfunc
 9: func m_vset(a;int)
10: str d[256]
11: int i,j
12: d="(v"+str$(a)+",0,"
13: for i=0 to 4:for j=0 to 10:d=d+str$(v(i,j))+",":next:next
14: d=left$(d,len(d)-1)+")"+chr$(13)+chr$(10)
15: fwrites(d,0)
16: endfunc
17: func m_alloc(t;char,s;int)
18: str d[256]
19: d="(m"+str$(t)+","+str$(s)+")"+chr$(13)+chr$(10)
20: fwrites(d,0)
21: endfunc
22: func m_assign(c;char,t;char)
23: str d[256]
24: d="(a"+str$(c)+","+str$(t)+")"+chr$(13)+chr$(10)
25: fwrites(d,0)
26: endfunc
27: func m_tempo(t;char)
28: str d[256]
29: d="(o"+str$(t)+")"+chr$(13)+chr$(10)
30: fwrites(d,0)
31: endfunc
32: func m_trk(t;char,s;str)
33: fwrites("(t"+str$(t)+")",0)
34: fwrites(s,0)
35: fwrites(chr$(13)+chr$(10),0)
36: endfunc
37: func m_play()
38: fwrites("(p)",0)
39: endfunc()
```

リスト4 サンプルミュージック(mp_name.bas)

```
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*     M A P P Y   N A M E   E N T R Y    /*
  30 /*                (C) namco               /*
  40 /*         Arranged  by Z.Nishikawa       /*
  50 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  60 m_init()
  80 dim char melody(4,10)={   /*SOUND NUMBER 1
  90 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 100   56, 15,  0,  0,  0,  0,  0,  0,  0,  3, 0,
 110 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 120   31,  0,  0,  0, 15, 35,  2, 13,  0,  0, 0,
 130   31,  0,  0,  0, 15, 48,  2,  9,  1,  0, 0,
 140   31,  0,  0,  0, 15, 28,  2,  1,  0,  0, 0,
 150   28,  0,  0,  5,  8,  0,  1,  1,  0,  0, 0}
 160 m_vset(1,melody)
 170 dim char bass(4,10)={     /*SOUND NUMBER 2
 180 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 190   61, 15,  0,  0,  0,  0,  0,  0,  0,  3, 0,
 200 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 210   31,  5,  3,  3,  1, 23,  0,  1,  3,  0, 0,
 220   31,  8,  7,  5,  3,  0,  1,  1,  0,  0, 0,
 230   31,  8,  7,  5,  2,  0,  1,  2,  7,  0, 0,
 240   31,  8,  7,  5,  2,  0,  1,  1,  7,  0, 0}
 250 m_vset(2,bass)
 260 dim char harm(4,10)={     /*SOUND NUMBER 3
 270 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 280    6, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 290 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 300   31,  0,  0, 15,  0, 10,  0,  6,  3,  0, 0,
 310   17, 18, 11, 15,  0,  0,  1,  3,  0,  0, 0,
 320   15, 20,  8, 15,  3,  0,  1,  0,  3,  0, 0,
 330   16, 16, 11, 15,  0,  0,  1,  1,  7,  0, 0}
 340 m_vset(3,harm)
 350 /*
 360 str a[256],b[256],c[256],d[256]
 370 /*
 400 for i=1 to 8:m_alloc(i,1000):m_assign(i,i):n
ext
 430 /*
 440 m_tempo(130)
 450 a="L16agf+gbag+a<c>ba+b<dc>b<c fed+edc>bag8<
g8>g4
 460 b="L4fgab c8r.rd
 470 c="L8a<a>b<bc<c>d<d>frr2>q4b4<q1
 480 m_trk(1,"@1 o4 q4 v12 y48,0 p3"+a)
 490 m_trk(2,"@1 o4 q4 v12 y49,24p3"+a)
 500 m_trk(3,"@1 o4 q4 v11 y50,0 p3"+b)
 510 m_trk(4,"@1 o4 q4 v11 y51,24p3"+b)
 520 m_trk(5,"@1 o2 q1 v12 y52,0 p3"+c)
 530 m_trk(6,"@2 o2 q1 v10 y53,20p3"+c)
 540 for i=7 to 8:m_trk(i,"|:18r1:|"):next
 550 /*
 560 a="L16|:<de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>
a4< de-dc>a8g8r8<cef8f+8 gagec8>g8r|1<c>a8g4:|<e-d
8c4>
 570 b="L16|:aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
   aga<c>a8g8r8b<cd8d+8 egec>a8g8r|1<c>a8g4:|<c>a
8g4
 580 c="L8|:c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>
c c.<c16>c<c>c<cc>c >g.<g16>g<g|1>ggab<:|c<cc>c
 590 m_trk(1,a)
 600 m_trk(2,a)
 610 m_trk(3,b)
 620 m_trk(4,b)
 630 m_trk(5,c)
 640 m_trk(6,c)
 650 /*
 660 a="L16|:b<c>bag+8e8r8a8a4 b<c>bag+8e8r8a8a4
b<c>bag+8b8r8<c8c4 de-dc>a8a-8g8<g8g4
 670 b="L16|:g+ag+ag+8e8r8a8a4 g+ag+ag+8e8r8a8a4
g+ag+ag+8e8r8a8a4 aga<c>a8a-8g8<g8>g4
 680 c=">L8e.<e16>e<e>>a<a>a<a e.<e16>e<e>>a<a>a<
a e.<e16>e<e>>a<a>a<a g.<g16>g<g>ggab<
 690 m_trk(1,a)
 700 m_trk(2,a)
 710 m_trk(3,b)
 720 m_trk(4,b)
 730 m_trk(5,c)
 740 m_trk(6,c)
 750 /*
 760 a="L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4<
 de-dc>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
 770 b="L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
 aga<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
 780 c="L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c
c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>c
 790 m_trk(1,a)
 800 m_trk(2,a)
 810 m_trk(3,b)
 820 m_trk(4,b)
 830 m_trk(5,c)
 840 m_trk(6,c)
 850 /*
 860 a="r1r2r8b4.
 870 b="r1r2r4d+4
 880 c="r1r2r4.g8
 890 d="L32p3g16p0r16p3gp0rp3gp0rp3g16p0r16p3gp0r
p3gp0rp3g16p0r16|:6p3gp0r:|p3g16p0r16|:6p3gp0r:|p3
g16p0r16r4.
 900 m_trk(1,"@3 o3 q8 v11 y48,0 p3"+a)
 910 m_trk(2,"@3 o3 q8 v11 y49,24p3"+a)
 920 m_trk(3,"@3 o4 q8 v11 y50,0 p3"+b)
 930 m_trk(4,"@3 o4 q8 v11 y51,24p3"+b)
 940 m_trk(5,"@3 o4 q8 v11 y52,0 p3"+c)
 950 m_trk(6,"@3 o4 q8 v11 y53,24p3"+c)
 960 m_trk(7,"v0@1 o1 q1 v12 y54,0 "+d)
 970 m_trk(8,"v0@2 o2 q1 v10 y55,0 "+d)
 980 /*
 990 a="L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4<
 de-dc>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
1000 b="L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
 aga<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
1010 c="L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c
c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>c
1020 m_trk(1,"@1 o5 q4 v12 y48,0 p3"+a)
1030 m_trk(2,"@1 o5 q4 v12 y49,24p3"+a)
1040 m_trk(3,"@1 o4 q4 v11 y50,0 p3"+b)
1050 m_trk(4,"@1 o4 q4 v11 y51,24p3"+b)
1060 m_trk(5,"@1 o3 q1 v12 y52,0 p3"+c)
1070 m_trk(6,"@2 o3 q1 v10 y53,20p3"+c)
1080 m_play()
```

サンプル曲，マッピーよりネームエントリー ©NAMCO

リスト5 BASIC版対応の修正(mpnm_cnv.bas)

```
   1 o_open()                       /*この１行を追加！
  10 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  20 /*     M A P P Y   N A M E   E N T R Y    /*
  30 /*                (C) namco               /*
  40 /*         Arranged  by Z.Nishikawa       /*
  50 /*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
  60 m_init()
  80 dim char v(4,10)={    /*配列変数はvにする
  90 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 100   56, 15,  0,  0,  0,  0,  0,  0,  0,  3, 0,
 110 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 120   31,  0,  0,  0, 15, 35,  2, 13,  0,  0, 0,
 130   31,  0,  0,  0, 15, 48,  2,  9,  1,  0, 0,
 140   31,  0,  0,  0, 15, 28,  2,  1,  0,  0, 0,
 150   28,  0,  0,  5,  8,  0,  1,  1,  0,  0, 0}
 160 m_vset(1)                 /*配列変数の引数をカット
 170 v={                       /*ここも同様に…
 180 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 190   61, 15,  0,  0,  0,  0,  0,  0,  0,  3, 0,
 200 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 210   31,  5,  3,  3,  1, 23,  0,  1,  3,  0, 0,
 220   31,  8,  7,  5,  3,  0,  1,  1,  0,  0, 0,
 230   31,  8,  7,  5,  2,  0,  1,  2,  7,  0, 0,
 240   31,  8,  7,  5,  2,  0,  1,  1,  7,  0, 0}
 250 m_vset(2)                 /*配列変数の引数をカット
 260 v={                       /*ここもvにする…
 270 /*AF   OM   WF   SY SP   PMD AMD PMS AMS PAN
 280    6, 15,  0,  0,  0,  0,  0,  0,  0,  3, 0,
 290 /*AR   DR   SR   RR   SL   OL   KS   ML DT1 DT2 AME
 300   31,  0,  0, 15,  0, 10,  0,  6,  3,  0, 0,
 310   17, 18, 11, 15,  0,  0,  1,  3,  0,  0, 0,
 320   15, 20,  8, 15,  3,  0,  1,  0,  3,  0, 0,
 330   16, 16, 11, 15,  0,  0,  1,  1,  7,  0, 0}
 340 m_vset(3)                 /*配列変数の引数をカット
 350 /*
 360 str a[256],b[256],c[256],d[256]
 370 /*
 400 for i=1 to 8:m_alloc(i,1000):m_assign(i,i):n
ext
 430 /*
 440 m_tempo(130)
 450 a="L16agf+gbag+a<c>ba+b<dc>b<c fed+edc>bag8<
g8>g4
 460 b="L4fgab c8r.rd
 470 c="L8a<a>b<bc<c>d<d>frr2>q4b4<q1
 480 m_trk(1,"@1 o4 q4 v12 y48,0 p3"+a)
 490 m_trk(2,"@1 o4 q4 v12 y49,24p3"+a)
 500 m_trk(3,"@1 o4 q4 v11 y50,0 p3"+b)
 510 m_trk(4,"@1 o4 q4 v11 y51,24p3"+b)
 520 m_trk(5,"@1 o2 q1 v12 y52,0 p3"+c)
 530 m_trk(6,"@2 o2 q1 v10 y53,20p3"+c)
 540 for i=7 to 8:m_trk(i,"|:18r1:|"):next
 550 /*
 560 a="L16|:<de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>
a4< de-dc>a8g8r8<cef8f+8 gagec8>g8r|1<c>a8g4:|<e-d
8c4>
 570 b="L16|:aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
   aga<c>a8g8r8b<cd8d+8 egec>a8g8r|1<c>a8g4:|<c>a
8g4
 580 c="L8|:c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>
c c.<c16>c<c>c<cc>c >g.<g16>g<g|1>ggab<:|c<cc>c
 590 m_trk(1,a)
 600 m_trk(2,a)
 610 m_trk(3,b)
 620 m_trk(4,b)
 630 m_trk(5,c)
 640 m_trk(6,c)
 650 /*
 660 a="L16|:b<c>bag+8e8r8a8a4 b<c>bag+8e8r8a8a4
b<c>bag+8b8r8<c8c4 de-dc>a8a-8g8<g8g4
 670 b="L16|:g+ag+ag+8e8r8a8a4 g+ag+ag+8e8r8a8a4
g+ag+ag+8e8r8a8a4 aga<c>a8a-8g8<g8>g4
 680 c=">L8e.<e16>e<e>>a<a>a<a e.<e16>e<e>>a<a>a<
a e.<e16>e<e>>a<a>a<a g.<g16>g<g>ggab<
 690 m_trk(1,a)
 700 m_trk(2,a)
 710 m_trk(3,b)
 720 m_trk(4,b)
 730 m_trk(5,c)
 740 m_trk(6,c)
 750 /*
 760 a="L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4<
 de-dc>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
 770 b="L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
 aga<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
 780 c="L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c
c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>c
 790 m_trk(1,a)
 800 m_trk(2,a)
 810 m_trk(3,b)
 820 m_trk(4,b)
 830 m_trk(5,c)
 840 m_trk(6,c)
 850 /*
 860 a="r1r2r8b4.
 870 b="r1r2r4d+4
 880 c="r1r2r4.g8
 890 d="L32p3g16p0r16p3gp0rp3gp0rp3g16p0r16p3gp0r
p3gp0rp3g16p0r16|:6p3gp0r:|p3g16p0r16|:6p3gp0r:|p3
g16p0r16r4.
 900 m_trk(1,"@3 o3 q8 v11 y48,0 p3"+a)
 910 m_trk(2,"@3 o3 q8 v11 y49,24p3"+a)
 920 m_trk(3,"@3 o4 q8 v11 y50,0 p3"+b)
 930 m_trk(4,"@3 o4 q8 v11 y51,24p3"+b)
 940 m_trk(5,"@3 o4 q8 v11 y52,0 p3"+c)
 950 m_trk(6,"@3 o4 q8 v11 y53,24p3"+c)
 960 m_trk(7,"v0@1 o1 q1 v12 y54,0 "+d)
 970 m_trk(8,"v0@2 o2 q1 v10 y55,0 "+d)
 980 /*
 990 a="L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4<
 de-dc>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
1000 b="L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4
 aga<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
1010 c="L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c
c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>c
1020 m_trk(1,"@1 o5 q4 v12 y48,0 p3"+a)
1030 m_trk(2,"@1 o5 q4 v12 y49,24p3"+a)
1040 m_trk(3,"@1 o4 q4 v11 y50,0 p3"+b)
1050 m_trk(4,"@1 o4 q4 v11 y51,24p3"+b)
1060 m_trk(5,"@1 o3 q1 v12 y52,0 p3"+c)
1070 m_trk(6,"@2 o3 q1 v10 y53,20p3"+c)
1080 m_play():end          /*なるべく「m_play」のうし
ろに「end」を付けよう!!
```

リスト6 OPMファイル(mp_name.opm)

```
(i)
(v1,0,56,15,0,0,0,0,0,0,0,3,0,31,0,0,0,15,35,2,13,
0,0,0,31,0,0,0,15,48,2,9,1,0,0,31,0,0,0,15,28,2,1,
0,0,0,28,0,0,5,8,0,1,1,0,0,0)
(v2,0,61,15,0,0,0,0,0,0,0,3,0,31,5,3,3,1,23,0,1,3,
0,0,31,8,7,5,3,0,1,1,0,0,0,31,8,7,5,2,0,1,2,7,0,0,
31,8,7,5,2,0,1,1,7,0,0)
(v3,0,6,15,0,0,0,0,0,0,0,3,0,31,0,0,15,0,10,0,6,3,
0,0,17,18,11,15,0,0,1,3,0,0,0,15,20,8,15,3,0,1,0,3
,0,0,16,16,11,15,0,0,1,1,7,0,0)
(m1,1000)
(a1,1)
(m2,1000)
(a2,2)
(m3,1000)
(a3,3)
(m4,1000)
(a4,4)
(m5,1000)
(a5,5)
(m6,1000)
(a6,6)
(m7,1000)
(a7,7)
(m8,1000)
(a8,8)
(o130)
(t1)@1 o4 q4 v12 y48,0 p3L16agf+gbag+a<c>ba+b<dc>b
<c fed+edc>bag8<g8>g4
(t2)@1 o4 q4 v12 y49,24p3L16agf+gbag+a<c>ba+b<dc>b
<c fed+edc>bag8<g8>g4
(t3)@1 o4 q4 v11 y50,0 p3L4fgab c8r.rd
(t4)@1 o4 q4 v11 y51,24p3L4fgab c8r.rd
(t5)@1 o2 q1 v12 y52,0 p3L8a<a>b<bc<c>d<d>frr2>q4b
4<q1
(t6)@2 o2 q1 v10 y53,20p3L8a<a>b<bc<c>d<d>frr2>q4b
4<q1
(t7)|:18r1:|
(t8)|:18r1:|
(t1)L16|:<de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4< d
e-dc>a8g8r8<cef8f+8 gagec8>g8r|1<c>a8g4:|<e-d8c4>
(t2)L16|:<de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4< d
e-dc>a8g8r8<cef8f+8 gagec8>g8r|1<c>a8g4:|<e-d8c4>
(t3)L16|:aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4    a
ga<c>a8g8r8b<cd8d+8 egec>a8g8r|1<c>a8g4:|<c>a8g4
(t4)L16|:aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4    a
ga<c>a8g8r8b<cd8d+8 egec>a8g8r|1<c>a8g4:|<c>a8g4
(t5)L8|:c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c c.<
c16>c<c>c<cc>c >g.<g16>g<g|1>ggab<:|c<cc>c
(t6)L8|:c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c c.<
c16>c<c>c<cc>c >g.<g16>g<g|1>ggab<:|c<cc>c
(t1)L16|:b<c>bag+8e8r8a8a4 b<c>bag+8e8r8a8a4 b<c>b
ag+8b8r8<c8c4 de-dc>a8a-8g8<g8g4
(t2)L16|:b<c>bag+8e8r8a8a4 b<c>bag+8e8r8a8a4 b<c>b
ag+8b8r8<c8c4 de-dc>a8a-8g8<g8g4
(t3)L16|:g+ag+ag+8e8r8a8a4 g+ag+ag+8e8r8a8a4 g+ag+
ag+8e8r8a8a4 aga<c>a8a-8g8<g8>g4
(t4)L16|:g+ag+ag+8e8r8a8a4 g+ag+ag+8e8r8a8a4 g+ag+
ag+8e8r8a8a4 aga<c>a8a-8g8<g8>g4
(t5)>L8e.<e16>e<e>>a<a>a<a e.<e16>e<e>>a<a>a<a e.<
e16>e<e>>a<a>a<a g.<g16>g<g>ggab<
(t6)>L8e.<e16>e<e>>a<a>a<a e.<e16>e<e>>a<a>a<a e.<
e16>e<e>>a<a>a<a g.<g16>g<g>ggab<
(t1)L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4< de-d
c>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
(t2)L16de-dc>a8g8r8<c8e-4 de-dc>a8g8r8<c8>a4< de-d
c>a8g8r8<cef8f+8 gagec8>g8r<e-d8c4>
(t3)L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4    aga
<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
(t4)L16aga<c>a8g8r8a8a4    aga<c>a8g8r8a8g4    aga
<c>a8g8r8b<cd8d+8 egec>a8g8r<c>a8g4
(t5)L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c c.<c1
6>c<c>c<cc>c >g.<g16>g<gc<cc>c
(t6)L8c.<c16>c<c>c<c>c<c>  c.<c16>c<c>c<cc>c c.<c1
6>c<c>c<cc>c >g.<g16>g<gc<cc>c
(t1)@3 o3 q8 v11 y48,0 p3r1r2r8b4.
(t2)@3 o3 q8 v11 y49,24p3r1r2r8b4.
(t3)@3 o4 q8 v11 y50,0 p3r1r2r4d+4
(t4)@3 o4 q8 v11 y51,24p3r1r2r4d+4
(t5)@3 o4 q8 v11 y52,0 p3r1r2r4.g8
(t6)@3 o4 q8 v11 y53,24p3r1r2r4.g8
(t7)@1 o1 q1 v12 y54,0 p0L32p3g16p0r16p3gp0rp3gp0r
p3g16p0r16p3gp0rp3gp0rp3g16p0r16|:6p3gp0r:|p3g16p0
r16|:6p3gp0r:|p3g16p0r16r4.
(t8)@2 o2 q1 v10 y55,0 p0L32p3g16p0r16p3gp0rp3gp0r
p3g16p0r16p3gp0rp3gp0rp3g16p0r16|:6p3gp0r:|p3g16p0
r16|:6p3gp0r:|p3g16p0r16r4.
(t1)@1 o5 q4 v12 y48,0 p3L16de-dc>a8g8r8<c8e-4 de-
dc>a8g8r8<c8>a4< de-dc>a8g8r8<cef8f+8 gagec8>g8r<e
-d8c4>
(t2)@1 o5 q4 v12 y49,24p3L16de-dc>a8g8r8<c8e-4 de-
dc>a8g8r8<c8>a4< de-dc>a8g8r8<cef8f+8 gagec8>g8r<e
-d8c4>
(t3)@1 o4 q4 v11 y50,0 p3L16aga<c>a8g8r8a8a4    ag
a<c>a8g8r8a8g4    aga<c>a8g8r8b<cd8d+8 egec>a8g8r<
c>a8g4
(t4)@1 o4 q4 v11 y51,24p3L16aga<c>a8g8r8a8a4    ag
a<c>a8g8r8a8g4    aga<c>a8g8r8b<cd8d+8 egec>a8g8r<
c>a8g4
(t5)@1 o3 q1 v12 y52,0 p3L8c.<c16>c<c>c<c>c<c>  c.
<c16>c<c>c<cc>c c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>
c
(t6)@2 o3 q1 v10 y53,20p3L8c.<c16>c<c>c<c>c<c>  c.
<c16>c<c>c<cc>c c.<c16>c<c>c<cc>c >g.<g16>g<gc<cc>
c
(p)
```

さあ，今月は半期に一度のMUSIC特集，当然LIVEもページ拡張さ！ てなことで，今回はOh!X LIVE SPECIAL版です。X68000用に3本，X1用に1本の計4本をどどどーっと掲載。ヘヴィメタあり，ゲームミュージックあり，はたまたイロモノありとバラエティ豊かに揃ったミュージックプログラムの数々。今回はリストもわりと短めなので，いままでものおじしていた人も打ち込んでみては？ 意外と秋の夜長にピッタリ合うかもよ。

X68000用 Rise And Fall

小野 智弘 Ono Tomohiro

7つの鍵を持て 開け地獄の門

LIVEスペシャルの1曲目は，HELLOWEENの“Keeper Of The Seven Keys Part 2”より“Rise And Fall”をお送りしましょう。HELLOWEENはカボチャのお面を付けて……失礼しました，ドイツのヘヴィメタルバンドです。このバンドについては，ハードロックだよっていう人もいますし，スラッシュに近いんでないのっていう人もいますが，とりあえずヘヴィメタルということにしておきましょう。メロディが美しく，その中にも激しさを取り入れた曲を数々聴かせてくれます。ジャーマンヘヴィメタルの頂点に君臨しているといっても過言ではないでしょう。

余談ですが，少年ジャンプに掲載されている（今は季刊）バスタードって漫画があるんですけど，ここに出てくる人名・地名・呪文名などは，ほとんどメタル関係で占められています。HELLOWEENもちゃんと登場しています。この作品をBGMにバスタードを読んでみたのですが，なかなか合っています。ちょうどエンドレスになっていますので，バスタードを持っている人は試してみてね。

リスト1 Rise And Fall

```
  10 /*
  20 /*  R i s e   a n d   f a l l -  H E L L O W E E N  -
  30 /*
  40 /*    From Keeper Of The Seven Keys pert2
  50 /*
  60 /*  P r o g r a m e d   b y   T o m o h i r o
  70 /*
  80 /*
  90 key 6,"width 96:m_stop()@M"
 100 key 7,"m_tempo(200)@M"
 110 key 8,"m_tempo(152)@M"
 120 /*
 130 str pd(99)[256]
 140 int po(256)
 150 /*                                              */
 160 /*         トラックバッファが足りないときは         */
 170 /*         CONFIG.SYSで，                        */
 180 /*               DEVICE = OPMDRV.X #128          */
 190 /*         のように指定すること                   */
 200 /*                                              */
 210 m_init()
 220 for k=1 to 8
 230   m_alloc(k,16000)
 240   m_assign(k,k)
 250 next
 260 m_tempo(152)
 270 /*
 280 /*
 290 str a,c,e,f,g,h,p1,p2,p3,s1,s2,s3
 300 str a1,a2,a3,a4,a5,a6,a7,a8,a9,a10
 310 /*
 320 vdata()
 330 ddata()
 340 pdata()
 350 m_play(1,2,3,4,5,6,7,8)
 360 end
 370 /*
 380 /*
 390 func trk(t)
 400   r=0
 410   while po(r)<>255
 420    m_trk(t,pd(po(r)))
 430     r=r+1
 440   endwhile
 450   return()
 460 endfunc
 470 /*
 480 /*
 490 func vdata()
 500 /*
 510 char guitar(4,10)={
 520             0, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 530            31,  2, 15,  3, 15, 18,  0,  0,  6,  0,  0,
 540            31,  3,  0,  3,  0, 38,  0,  0,  5,  0,  0,
 550            31,  2,  0,  3,  0, 16,  0,  2,  1,  0,  0,
 560            31,  1,  0,  3,  0,  0,  0,  0,  2,  0,  0}
 570 m_vset(69,guitar)
 580 /*
 590 char vocal(4,10)={
 600            60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 610            31,  0,  1,  4,  1, 25,  0,  2,  7,  0,  0,
 620            26,  2,  2,  4,  1,  0,  0,  1,  3,  0,  0,
 630            29,  0,  1,  4,  1, 30,  0,  3,  7,  0,  0,
 640            24,  2,  3,  4,  2,  0,  0,  2,  3,  0,  0}
 650 m_vset(70,vocal)
 660 /*
 670 char bass(4,10)={
 680             0, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 690            31, 15,  0,  6,  0, 16,  0,  5,  7,  0,  0,
 700            31, 11,  3,  6,  2, 38,  0,  1,  5,  0,  0,
 710            31, 10,  5,  6,  4, 19,  0,  0,  1,  0,  0,
 720            31,  8,  8,  6,  8,  0,  0,  0,  3,  0,  0}
 730 m_vset(71,bass)
 740 /*
 750 char bdrum(4,10)={
 760            59, 15,  0,  0,  0,  0,  0,  0.  0,  3,  0,
 770            31,  0,  0,  0,  0, 20,  0,  4,  0,  0,  0,
 780            31, 27,  0,  0, 15, 15,  0,  8,  0,  0,  0,
 790            31, 21,  0,  0, 15,  5,  0,  1,  0,  0,  0,
 800            31,  8,  0,  3, 15,  0,  3,  0,  0,  2,  0}
 810 m_vset(72,bdrum)
 820 /*
 830 char snare(4,10)={
 840            60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
 850            31,  0,  2, 15,  4,  0,  0,  1,  0,  3,  0,
 860            31, 13,  4, 15,  8,  0,  0,  1,  0,  0,  0,
 870            31, 31,  2, 15,  4,  0,  0,  5,  0,  0,  0,
 880            31, 20,  4, 15,  8,  0,  0,  1,  0,  0,  0}
 890 m_vset(73,snare)
 900 /*
 910 char hihat(4,10)={
 920            60, 15,  2,  0,  0,  0,  0,  0,  0,  3,  0,
 930            31, 15,  8, 15,  5,  0,  0,  8,  7,  1,  0,
 940            31, 20, 10, 15, 15,  0,  0,  4,  3,  0,  0,
 950            31, 15,  8, 15,  5,  0,  0,  8,  7,  1,  0,
 960            31, 20, 10, 15, 15,  0,  0,  2,  3,  0,  0}
 970 m_vset(74,hihat)
 980 /*
 990 char tom(4,10)={
1000            62, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
1010            31,  0,  1,  0,  2,  0,  0,  0,  7,  0,  0,
1020            31, 16, 10,  6, 15,  0,  0,  0,  5,  0,  0,
1030            31, 10,  8,  6, 12,  0,  0,  0,  6,  0,  0,
1040            31, 13,  6,  6,  4,  0,  0,  1,  1,  0,  0}
1050 m_vset(75,tom)
1060 /*
1070 char se(4,10)={
1080            60, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
1090            31, 10,  1,  4, 15, 35,  0,  8,  7,  0,  0,
1100            31,  0,  0,  4,  0,  0,  0,  1,  3,  0,  0,
```

```
1110               31, 10, 15,  4, 15, 37,  0,  1,  7,  0,  0,
1120               31,  0,  0,  4,  0,  0,  0,  1,  3,  0,  0}
1130 m_vset(76,se)
1140 /*
1150 char guitar2(4,10)={
1160                0, 15,  0,  0,  0,  0,  0,  0,  0,  3,  0,
1170               31,  2,  0,  0,  0, 19,  0,  0,  0,  0,  0,
1180               31,  4,  0,  2,  1, 30,  0,  1,  0,  0,  0,
1190               31,  2,  0,  0,  1, 14,  0,  2,  0,  0,  0,
1200               31,  4,  0,  2,  1,  0,  0,  0,  0,  0,  0}
1210 m_vset(77,guitar2)
1220 return()
1230 /*
1240 /*
1250 func ddata()
1260 /*
1270 a="y2,23a":c="@73y2,3a":e="@73y2,9e":f="@73y2,14e":g="@74y
2,65g":h="@74y2,8a
1280 p1="y3,1":p2="y3,2":p3="y3,3":s1="@75y2,25e":s2="@75y2,26a
":s3="@75y2,27b
1290 a1="e&d#&e&d#&":a2="d#&d&d#&d&":a3="d&c#&d&c#&":a4="c#&c&c
#&c&":a5="c&>b&a#&b&<
1300 a6="b&a#&a&a#&":a7="a&g#&g&g#&":a8="g&f#&f&f#&":a9="f&e&d#
&e&":a10="e&d#&d&d#&
1310 return()
1320 /*
1330 /*
1340 func pdata()
1350 /*
1360 /* vocal
1370 /*
1380 pd(97)="[d.c.]
1390 pd(98)="[coda]
1400 pd(99)="[*]
1410 pd(0)="y48,32r4r4
1420 pd(1)="r1r1r1r1
1430 pd(2)="|:r1r1:|
1440 pd(3)="r2r4@70v13p3o518rc#aa4v11rr2r1r4rre4e4
1450 pd(4)="@70v13o514a8ar8baf#f#>r8b8<f#8f#8
1460 pd(5)="e#8&c#c#8e#g#164f#8&f&e&d#&d&c#&c&>b&a#14v1rv13<f#g
#
1470 pd(6)="164a8&g#&g&f#&f&e&d#&d&c#14abaa8&f#8f#r8f#8&e#
1480 pd(7)="e#8&e#8c#<c#>b8a8164a8&g#&g&f#&f&e&d#&d&c#14v1rv13f
#g#
1490 pd(8)="aabaa8&f#8f#r8>a8<f#8&e#8
1500 pd(9)="e#8&c#8c#e#g#164f#4&f#8&f&e&d#&d&c#&c&>b&a#<14r8f#8
g#
1510 pd(10)="a8a8&abaaf#r8>f#8<f#
1520 pd(11)="e#e#f#g#8&f#8f#r<@76132a#&a&g#&g&f#&f&e&d#&d&c#&c&
>b&a#&a&g#>v1r@7014
1530 pd(12)="v14q6o4a8<a>a8<a>a8<a8>a8<af#8aa8a8
1540 pd(13)="o4b8<af#8ab8a8q8g#2&er8>a8
1550 pd(14)="q6o4a8<a>a8<a>a8<a8>a8<ag#8ag#8a8
1560 pd(15)="o4b8<af#8ab8a8g#rq8@v122ba
1570 pd(16)="o518b4<c#>a&a4r>a<b4<c#&>af#4f#f#
1580 pd(17)="14<c#d>b&a&f#8&g#.r418ba#
1590 pd(18)="b4<c#>a&164a8&g#&g&f#&f&e&d#&d&c#18r>a<b4<c#>a&f#4
r>b
1600 pd(19)="14<<c#d>b&a&164g#8&g&f#&f&e&d#&d&c#&c18v1r@v122>b<
g#g#f#e
1610 pd(20)="o514f#&e2rr1
1620 pd(21)="r1r418r>g#<f#164a4&g#&g&f#&f&e&d#&d&c#
1630 pd(22)="f#4.&e8&e2r1@77v15
1640 pd(94)="@69v15o6164"+a1+a1+a1+a1+a2+a2+a2+a2+a3+a3+a3+a3+a
4+a4+a4+a4
1650 pd(95)=a5+a5+a5+"o5"+a6+a6+a6+a7+a7+a7+a8+a8+a8+a9+a9+a10+
a10
1660 pd(96)="y48,32@70v1r2r4v12o4a#&b&<c&c#&d&d#&e&f&f#8
1670 po={97,0,1,2,3,98,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,1
9,20,21,
1680      4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,255}
1690 trk(1)
1700 /*
1710 /*
1720 /* e.g.1-1
1730 /*
1740 pd(2)="@69v15p1q7o3|:a2&a8v12116eeaaeev15f#2&f#8v12eef#f#e
e:|
1750 pd(3)="v15o314a.b8<{c#>ba}2f#.g#8{ag#e}2 a.b8<{c#>ba}2f#.g
#8{ag#e}2
1760 pd(4)="o3v15116aaaav1418aav13aav12aav15116<ddddv1418ddv13d
dv12dd
1770 pd(5)="v15o4116c#c#c#c#v1418c#c#v13c#c#v12c#c#v15116>f#f#f
#f#v14f#8f#f#v1518<c#>bag#
1780 pd(6)="v15o4116ddddv1418ddv11ddv13ddv15116>bbbbv1418bbv11b
bv12bb
1790 pd(7)="v15o4116c#c#c#c#v1418c#c#v13c#c#v12c#c#v15116>f#f#f
#f#v14f#8f#f#v1518ag#<c#>b
1800 pd(12)="v15o412f#.b4a.b4
1810 pd(13)="a.d4c#b
1820 pd(14)="o4116c#c#c#c#c#c#c#c#c#c#eeeef#f#f#f#f#f#f#f#f
#f#f#f#eeee
1830 pd(15)="ddddddddddddddeeeeeeeeeeeeeeee
1840 pd(16)="o318a2.ag#f#2.f#f#
1850 pd(17)="<d2.d#4e2f2
1860 pd(18)="f#2.f#ed2.dc#
1870 pd(19)=">b2.<c4c#2e2
1880 pd(20)="o514c#.d8{edc#}2>a.b8<{c#>bg#}2
1890 pd(22)="o3164a2.&a8&g#&g&f#&f&e&d#&d&c#<a2.&a8&g#&g&f#&f&e
&d#&d&c#
1900 pd(23)="q2o5116c#c#eeddc#c#>f#f#g#g#aa<c#c#>e#e#f#f#g#g#bb
aaf#f#eec#c#
1910 pd(24)="f#f#g#g#aa<c#c#>ddeef#f#aae#e#f#f#g#g#bbaaf#f#eedd
1920 pd(25)="c#c#>bbaag#g#p1
1930 pd(26)="v15o5q7164c#2&c#8&c&>b&a#&a&g#&a&a#&b&<116c#&c&c#&
d&c#&c&c#&d&d&c#&d&d#&d&c#&d&d#&d&c#&d&d#
1940 pd(27)="d&c#&d&d#>af#a8&a4&a#8.116&ba32&a#32&bf#8&f#e#&f#&
e#{c#e#f#}4>b&<c&>b32&a#32&a32
1950 pd(28)="<e#&f#8.<f#4&132f#&f&f#&f&f#&f&f#&f&f#&f&f#&f116c#
d#164f#16&f#&g&g#&a&a#&b&<c&c#&d&d#&e&f&f#4e8d#8{e&dc#}16116rc#c
>b
1960 pd(29)="a8b8<c#8.>baarf#r132a&a#&b&a#&a&a218>f#ab<c#&
1970 pd(30)="o312a.g#4f#.e4
1980 pd(31)="<d.c#4>b<c#
1990 pd(34)=">a1&164g#2&g4&f#8&f&e&d#&d&c#&c&>b&a#v0
2000 pd(35)="v15o418rc#>b<c#def#g#
2010 pd(38)="o412e.b4a.b4
2020 pd(39)="a.c4bb
2030 po={23,24,25,255}:trk(1):po={26,27,28,29,255}:m_trk(1,"p2"
):trk(1)
2040 po={97,0,1,2,3,98,4,5,6,7,4,5,6,5,12,13,14,15,16,17,18,19,
20,20,
2050      4,5,6,7,4,5,6,5,12,13,38,39,16,17,18,19,22,
2060      23,24,25,26,27,28,29,30,31,30,31,34,35,99,255}
2070 m_trk(2,"y49,32"):trk(2)
2080 /*
2090 /*
2100 /* e.g.1-2
2110 /*
2120 pd(2)="@69v15p1q6o4|:e2&e8116>eeaaee<c#2&c#8>eef#f#ee:|
2130 pd(3)="o314a.b8<{c#>ba}2f#.g#8{ag#e}2 o5c#.d8{edc#}2>a.b8{
<c#de}2
2140 pd(12)="v15o412a.<e4d.e4
2150 pd(13)="d.c#4>b<e
2160 pd(16)="o418e2.ed#c#2.c#c#
2170 pd(17)="a2.a#4b2<c2
2180 pd(18)="c#2.c#>ba2.ag#
2190 pd(19)="f#2.g4g#2b2
2200 pd(20)="o514a.b8<{c#>ba}2f#.g#8{ag#e}2
2210 pd(22)="o4164e2.&e8&d#&d&c#&c&>b&a#&a&g#a2.&a8&g#&g&f#&f&e
&d#&d&c#
2220 pd(30)="o412e.d#4c#.>b4
2230 pd(31)="<a.g#4f#g#
2240 pd(34)="e2&d#2&164d2&c#4&c8&>b&a#&a&g#&g&f#&f&ev0
2250 pd(38)="o412a.<e4d.e4
2260 pd(39)="d.>g4<ee
2270 m_trk(3,"y50,64r64"):trk(3)
2280 /*
2290 /*
2300 /* e.g.2-1
2310 /*
2320 pd(2)="@69v15p2q7o3|:a2&a8v12116eeaaeev15f#2&f#8v12eef#f#e
e:|
2330 pd(3)="o3|:a8116aaa8aaa8eeaaeef#8f#f#f#8f#f#f#8eef#f#ee:|
2340 pd(5)="v15o4116c#c#c#c#v1418c#c#v13c#c#v12c#c#v15116>f#f#f
#f#v14f#8f#f#v1518ag#f#e
2350 pd(7)="v15o4116c#c#c#c#v1418c#c#v13c#c#v12c#c#v15116>f#f#f
#f#v14f#8f#f#v1518f#eag#
2360 pd(12)="v15o3116aaaaaaaaaaaa<c#c#c#c#ddddddddddddc#c#c#c#
2370 pd(13)="o3bbbbbbbbbbbb<ccccc#c#c#c#c#c#c#eeeeeeee
2380 pd(16)="o318a2.ag#f#2.f#f#
2390 pd(17)="<d2.d#4e2f2
2400 pd(18)="f#2.f#ed2.dc#
2410 pd(19)=">b2.<c4c#2e2
2420 pd(22)="o3164a1&a2.&a8&g#&g&f#&f&e&d#&d&c#
2430 pd(23)="q2o4116aa<c#c#>bbaaddeef#f#aac#c#d#d#e#e#g#g#f#f#d
dc#c#>bb
2440 pd(24)="<ddeef#f#aa>bb<c#c#ddf#f#c#c#d#d#e#e#g#g#f#f#ddc#c
#>bb
2450 pd(25)="aag#g#f#f#ee
2460 pd(26)="v15p2q7o312a.g#4f#.e4
2470 pd(27)="<d.c#4>b<c#
2480 pd(30)="o5r8116aag#f#eedc#r8>baaa132b&<c&c#&c&>b&<c&c#16&c
#&c&>b16a8116f#&f&f#&fr4
2490 pd(31)="o6{f#g#a&g#f# g#g#a&g#f# aab&ag# abag#a}1<e4.&d#8>
b4.aa
2500 pd(32)="b<c#>ba b<c#>ba b<c#>ba {b<c#>bag#}4 abag# abag# a
bag# abag#
2510 pd(33)="f#g#ag# {f#g#ag#f#}4{g#g#ag#f#}4{g#af#ee}4f#&f&132
e&f&f#1618dc#>baf#a
2520 pd(34)=">a1&164g#2&g4&f#8&f&e&d#&d&c#&c&>b&a#v0
2530 pd(35)="v15o318rag#ab<c#de
2540 pd(36)="o312a.<c#4d.c#4
2550 pd(37)=">b.<c4c#e
2560 po={30,31,32,33,94,95,96,99,255}:m_trk(1,"y48,48p1"):trk(1
)
2570 po={97,0,1,2,3,98,4,5,6,7,4,5,6,5,12,13,12,13,16,17,18,19,
3,
2580      4,5,6,7,4,5,6,5,36,37,36,37,16,17,18,19,22,
2590      23,24,25,26,27,26,27,30,31,32,33,34,35,99,255}
2600 m_trk(4,"y51,48r64"):trk(4)
2610 /*
2620 /*
2630 /* e.g.2-2
2640 /*
2650 pd(2)="@69v15p2q6o4|:e2&e8116>eeaaee<c#2&c#8>eef#f#ee:|
2660 pd(3)="o4|:e8116eee8eee8>eeaaee<c#8c#c#c#8c#c#c#8>eef#f#ee
:|
2670 pd(11)="v15o4116c#c#c#c#v1418c#c#v13c#c#v12c#c#v15116>f#f#
f#f#v14f#8f#f#v1518ag#f#
2680 pd(12)="y52,32@70v12p3q6o518c#c#f#4c#f#4c#f#df#4df#4f#f#
2690 pd(13)="dd4dd4ddq8e2&>b4r<q6c#
2700 pd(14)="c#e4c#e4c#edf#4df#4dd
2710 pd(15)="dd4dd4dde4r2.
2720 pd(16)="y52,80@69v15p2q6o418e2.ed#c#2.c#c#
2730 pd(17)="a2.a#4b2<c2
2740 pd(18)="c#2.c#>ba2.ag#
2750 pd(19)="f#2.g4g#2b2
2760 pd(22)="o4164e1&e2.&e8&d#&d&c#&c&>b&a#&a&g#
2770 pd(26)="v15p2q7o412e.d#4c#.>b4
2780 pd(27)="<a.g#4f#g#
2790 pd(34)="e2&d#2&164d2&c#4&c8&>b&a#&a&g#&g&f#&f&ev0
2800 pd(36)="o412e.g#4a.g#4
2810 pd(37)="f#.g4g#b
2820 po={97,0,1,2,3,98,4,5,6,7,4,5,6,11,12,13,14,15,16,17,18,19
,3,
2830      4,5,6,7,4,5,6,11,12,13,14,15,16,17,18,19,22,
2840      23,24,25,26,27,26,27,30,31,32,33,34,35,99,255}
2850 m_trk(5,"y52,80"):trk(5)
2860 /*
2870 /*
2880 /* e.b.
2890 /*
2900 pd(2)="@71v15p3o2|:a2&a8116eeaaeef#2&f#8eef#f#ee:|
2910 pd(3)="o2|:a8116aaa8aaa8eeaaeef#8f#f#f#8f#f#f#8eef#f#ee:|
2920 pd(4)="o214aaaa8<c#8dddd
2930 pd(5)="o3c#c#c#8c#8c#8&>b8f#f#a8g#8f#8<c#8
2940 pd(6)="o3dddd8c#8>bbbb8b8
2950 pd(7)="o3c#c#c#8>b8a8g#8f#f#g#8f#8a8g#8
2960 pd(8)="o2aaaa<dddd
```

```
2970 pd(9)="o3c#c#c#8>b8a8g#8f#f#a8g#8f#8<c#8
2980 pd(10)="o3ddd8>a8<d8c#8>bbbb8b8
2990 pd(11)="o3c#c#c#8>b8a8g#8f#f#a8g#8f#8e8
3000 pd(12)="o218arr4r4<c#rdrr4r4c#r
3010 pd(13)=">brr4r4crc#rr4e2
3020 pd(14)="o2arr4r4<c#rdrr4r4c#r
3030 pd(15)=">brr4r4<crc#rr4>e2
3040 pd(16)="o218a<c#ag#f#edc#>f#<c#>f#<c#>f#<c#>f#<c#
3050 pd(17)="df#<dc#>bag#f#ebebfbfb
3060 pd(18)="f#<c#dc#>bag#ada16.&a#32&bag#f#ef#
3070 pd(19)="g#ab<c#>bag#ac#g#c#g#>e2
3080 pd(22)="o2a2&164a4.&g#&g&f#&f&e&d#&d&c#d2.&d8&c#&c&>b&a#&a
&g#&g&f#
3090 pd(23)="o2116aaaaaaaa<ddddddddc#c#c#c#c#c#c#>f#f#f#f#f
#f#f#
3100 pd(24)="<dddddddd>bbbbbbbb<c#c#c#c#c#c#c#f#f#ddc#c#>bb
3110 pd(25)="aag#g#f#f#ee
3120 pd(26)="o2116aaaaaaaaaaaag#g#g#g#f#f#f#f#f#f#f#f#f#f#e
eee
3130 pd(27)="<dddddddddddc#c#c#c#>bbbbbbbb<c#c#c#c#>eeee
3140 pd(34)="o218y53,32a&y53,28a&y53,24a&y53,20a&y53,16a&y53,12
a&y53,8a&y53,4a&y53,32g#4r4r2
3150 pd(35)="o218rag#ab<c#de
3160 pd(36)="o218aaaaaa<c#c#dddddc#c#
3170 pd(37)=">bbbbbb<ccc#c#c#c#e4>e4
3180 po={97,0,1,2,3,98,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,1
9,3,
3190      4,5,6,7,8,9,10,11,36,37,36,37,16,17,18,19,22,
3200      23,24,25,26,27,26,27,26,27,26,27,34,35,99,255}
3210 m_trk(6,"y53,32"):trk(6)
3220 /*
3230 /*
3240 /* b.d.
3250 /*
3260 pd(1)="@72v15p2o214"+p3+a+a+a+"8.116"+a+a+a+a+a
3270 pd(2)="14"+a+a+a+"r"
3280 pd(3)=a+"8116"+a+a+"r4r4"+a+a+a+a
3290 pd(4)=a+"8116"+a+a+"r2.
3300 pd(5)="116"+a+a+a+a+"18r"+a+a+a+"r"+a
3310 pd(6)="116"+a+a+a+a+"18r"+a+"14"+a+a
3320 pd(7)="116"+a+a+a+a+"r8"+a+a+a+"4"+a+a+a+"8
3330 pd(8)="14"+a+"rr"+a
3340 pd(9)="14"+a+"r"+a+"r
3350 pd(10)=a+"4r2.
3360 pd(11)="116"+a+a+a+a+a+a+a+a+a+a+a+a+a+a+a+a
3370 pd(12)="116"+a+a+a+a+a+a+a+a+a+a+a+a+a+a+a+"8
3380 pd(13)="116"+a+a+a+a+a+a+a+a
3390 pd(14)="14"+a+a+a+a
3400 pd(15)="18r"+a+a+a+a+a+a+a
3410 pd(16)=a+"418r"+a+"r2
3420 po={97,0,1,1,1,2,1,1,1,1,3,3,3,4,98,5,5,5,6,5,5,5,6,5,5,5,
6,5,5,5,7,
3430      8,8,8,9,8,8,8,10,11,11,11,11,11,11,11,12,3,3,3,4,
3440      5,5,5,6,5,5,5,6,5,5,5,6,5,5,5,7,
3450      14,14,14,14,14,14,14,16,11,11,11,11,11,11,11,12,11,10,
11,11,11,11,13,
3460      11,11,11,11,11,11,11,11,11,11,11,11,11,11,11,12,14,14,
15,99,255}
3470 m_trk(7,"y54,0"):trk(7)
3480 /*
3490 /*
3500 /* s.d. & hihat & cymbal & tom & clap
3510 /*
3520 pd(0)="v15p2o218r"+p3+c+"4"+g
3530 pd(1)="o218"+g+e+g+e+g+e+g+f
3540 pd(2)="18"+g+e+g+e+"116"+g+e+e+e+e+e+e+e
3550 pd(3)="14"+g+c+c+"18"+g+g
3560 pd(4)="14"+g+c+c+"8116"+e+e+e+e+e+e
3570 pd(5)="18"+g+g+e+g+g+g+e+g
3580 pd(6)="18"+g+g+e+g+g+f+"4"+f
3590 pd(7)="18"+g+g+e+g+g+f+"4"+g
3600 pd(8)="14"+e+"rr"+e
3610 pd(9)="14"+e+"r"+c+"2
3620 pd(10)="18"+g+p2+s1+p3+e+"16"+e+"."+p2+s1+".116"+p3+e+"r"+
e+e+e
3630 pd(11)="18"+g+e+"4"+e+g+e+g+f
3640 pd(12)="18"+g+e+g+e+g+e+"116"+p2+h+h+p1+h+h+p3
3650 pd(13)="18"+e+g+"4"+g+g+e+g+e
3660 pd(14)=g+"8116"+e+e+e+e+e+e+p2+s2+s2+s2+s2+p1+s3+s3+s3+s3+
p3
3670 pd(15)="18"+g+e+g+"116"+p2+h+p1+h+p3
3680 pd(16)="18"+g+e+g+"116"+g+p2+s2+s3+"8116"+s3+s3+p1+s1+s1+s
1+s1+p3
3690 pd(17)="18"+g+p2+s3+"116"+p3+e+p1+s3+s3+s3+p2+s2+p1+s3+s3+
s3+p2+s1+"8"+p1+s3+s3
3700 pd(18)="116"+p2+s2+p1+s3+s3+s3+p2+s2+p1+s3+s3+s3+p2+s2+p1+
s3+s3+s3+p3+c+"4
3710 pd(19)="18r"+g+g+g+g+g+g+g
3720 pd(20)=g+"418"+e+g+g+g+e+g
3730 pd(21)=g+"418"+e+g+g+"4"+e+"116"+p2+h+p1+h+p3
3740 pd(22)=g+"418"+e+g+g+g+e+"4
3750 pd(23)="18"+g+e+g+g+e+"116"+e+e+e+e+e+e
3760 po={97,0,1,1,1,2,1,1,1,1,3,3,3,4,98,5,5,5,6,5,5,5,6,5,5,5,
6,5,5,5,7,
3770      8,8,8,9,8,8,8,10,11,1,1,12,13,1,1,1,3,3,3,4,
3780      5,5,5,6,5,5,5,6,5,5,5,6,5,5,5,7,
3790      20,20,20,21,22,22,20,23,11,1,1,12,13,1,1,1,1,14,1,1,1,
1,15,
3800      1,1,1,12,1,1,1,16,11,1,1,1,11,1,1,1,17,18,19,99,255}
3810 m_trk(8,"y55,32"):trk(8)
3820 /*
3830 return()
```

X68000用 NINJA WARRIORSより PARADOX

西本 英樹 Nishimoto Hideki

久し振りのZUNTATAだよん

NINJA WARRIORSより，“PARADOX”をお送りしましょう。TAITOゲームミュージックの極みともいえるNINJA WARRIORSでは，業界初の津軽三味線を使用していた“DUDDY MULK”に目を（耳を）奪われがちですが，“ARE YOU LADY”やこの“PARADOX”のような名曲もあるんですよ。作ってくれたのは，すっかり常連入りした西本君ですね。

曲のほうのデキはよいのですが，ボスコニアンのサンプリングと曲との相性が，あまりよくないようで残念です。同封のディスクにはオリジナルのサンプリングデータが入っていましたが，さすがに掲載することはできませんでした。ちなみに，西本君

ちなみに，西本君は気づいていたのかもしれませんが，Oh!X LIVEのリストは横幅が64文字で打ち出されています。REM文を入れるときはWIDTH 64にして画面に入るようにすると，掲載されたときにきれいになりますよ。

リスト2 PARADOX

```
 10 /*
 20 /*         T h e   N I N J A   W A R R I O U R S
 30 /*
 40 /*        [ P A R A D O X ( E n d i n g   T h e m e ) ]
 50 /*
 60 /*         ( C ) T A I T O  MCMLXXXVIII / Z U N T A T A
 70 /*
 80 /*         P r o g r a m   a r r a n g e d   b y
 90 /*
100 /*         H I D E K I   -   N I S H I M O T O .
110 /*
120 m_init()
130 /*
140 for Z=1 to 8
150       m_assign(Z,Z)
160       m_alloc(Z,1400)
170 next
180 /*
190 str B1[32],B2[32],B3[32]
200 str S1[32],S2[32],S3[32]
210 str R1[32],R2[32],R3[32]
220 str  B[32], S[32],CL[32]
230 str  H[32], M[32], L[32]
240 str  C[32], O[32]
250 str ZA[256],ZB[256]
260 dim str X(64)[256]
270 /*
280 m_tempo(96)
290 /*
300 /*         V o i c e   D a t a
310 /*
320 /*                                   F a d e   I n
330 dim char V(4,10)={
340 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
350       60, 15,  2,  0,206, 16,  0,  5,  0,  3,  0,
360 /*    AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
370        2,  0,  0,  7,  0, 38,  0,  2,  0,  0,  0,
380        2,  0,  0,  7,  0, 12,  0,  2,  0,  0,  0,
390        2,  0,  0,  7,  0, 35,  0,  2,  0,  0,  0,
400        2,  0,  0,  7,  0,  6,  0,  2,  0,  0,  0}
410 m_vset(1,V)
420 /*                                   A r p e d i o
430 V={
440 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
450       51, 15,  2,  0,206, 16,  0,  1,  0,  3,  0,
460 /*    AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
470       31, 14,  0,  7, 12, 49,  1, 10,  0,  0,  0,
480       31, 15,  0,  6, 14, 57,  2,  9,  0,  0,  0,
490       31, 10,  0,  6,  0, 69,  1,  3,  3,  0,  0,
500       31, 10,  0,  5, 13,  0,  2,  1,  7,  0,  0}
510 m_vset(2,V)
520 /*                                   S y n t h I
```

```
530 V={
540 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
550       28, 15,  2,  0,206, 16,  0,  5,  0,  3,  0,
560 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
570       31,  0,  8,  3,  4, 26,  2,  2,  3,  0,  0,
580       31,  8,  6, 10,  4,  0,  2,  4,  3,  0,  0,
590       31,  0,  8,  3,  4, 26,  2, 12,  7,  0,  0,
600       31,  8,  6, 10,  4,  0,  2,  2,  7,  0,  0}
610 m_vset(3,V)
620 /*                                  S y n t h  II
630 V={
640 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
650       59, 15,  2,  0,206, 16,  0,  1,  0,  3,  0,
660 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
670       31, 10,  8,  7,  8, 56,  0,  0,  3,  1,  0,
680       31, 15,  0,  7,  6, 32,  0,  8,  7,  0,  0,
690       31, 10,  9,  7,  6, 52,  0,  0,  0,  0,  0,
700       31, 10,  9,  7,  6,  0,  0,  1,  0,  0,  0}
710 m_vset(4,V)
720 /*                                  S y n t h  III
730 V={
740 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
750       62, 15,  2,  0,206, 16,  0,  1,  0,  3,  0,
760 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
770       31,  5,  8,  5,  2, 42,  0, 11,  0,  2,  0,
780       31,  6, 11,  5,  1,  0,  0,  1,  0,  0,  0,
790       31,  6,  8,  5,  2,  7,  0,  1,  0,  0,  0,
800       31,  8,  9,  5,  1,  4,  0,  5,  0,  1,  0}
810 m_vset(5,V)
820 /*                                  B a s s  I
830 V={
840 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
850       34, 15,  2,  0,206, 16,  0,  1,  0,  3,  0,
860 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
870       31,  0,  0,  7,  0, 24,  0,  1,  1,  0,  0,
880       31,  0,  0,  7,  0, 63,  0,  0,  3,  0,  0,
890       31,  0,  0,  7,  0, 23,  0,  0,  5,  0,  0,
900       31,  0,  0,  7,  0,  0,  0,  0,  7,  0,  0}
910 m_vset(6,V)
920 /*                                  B a s s  II
930 V={
940 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
950       58, 15,  2,  0,206, 16,  0,  1,  0,  3,  0,
960 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
970       31, 10,  7,  8,  2, 33,  0,  0,  7,  0,  0,
980       31,  8,  8,  7,  5, 23,  3,  7,  7,  2,  0,
990       31,  5,  6,  7,  1, 37,  0,  0,  3,  0,  0,
1000      31,  8,  6,  6,  5,  2,  0,  0,  3,  0,  0}
1010 m_vset(7,V)
1020 /*                                 C h o r u s  I
1030 V={
1040 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
1050      12, 15,  2,  0,206, 16,  0,  5,  0,  3,  0,
1060 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
1070      12,  2,  0, 11,  1, 42,  0,  8,  3,  0,  0,
1080      12,  2,  0, 11,  1,  0,  0,  8,  3,  0,  0,
1090      12,  2,  0, 11,  1, 50,  0,  8,  7,  0,  0,
1100      12,  2,  0, 11,  1,  0,  0,  8,  7,  0,  0}
1110 m_vset(8,V)
1120 /*                                 C h o r u s  II
1130 V={
1140 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
1150       4, 15,  2,  0,206, 16,  0,  5,  0,  3,  0,
1160 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
1170      12,  2,  0, 11,  1, 42,  0,  4,  3,  0,  0,
1180      12,  2,  0, 11,  1,  0,  0,  4,  3,  0,  0,
1190      12,  2,  0, 11,  1, 50,  0,  4,  7,  0,  0,
1200      12,  2,  0, 11,  1,  0,  0,  4,  7,  0,  0}
1210 m_vset(9,V)
1220 /*                                 C h o r u s  III
1230 V={
1240 /*  F/A  OM  WF SYC SPD PMD AMD PMS AMS PAN
1250      36, 15,  2,  0,206, 16,  0,  5,  0,  3,  0,
1260 /*   AR D1R D2R  RR D1L  TL  KS MUL DT1 DT2 AMSE
1270      20,  6,  2,  2,  2, 44,  0,  8,  7,  0,  0,
1280      14,  5,  3,  9,  3,  2,  0,  8,  7,  0,  0,
1290      20,  6,  2,  2,  2, 45,  0,  8,  3,  0,  0,
1300      14,  5,  3,  9,  3,  1,  0,  8,  3,  0,  0}
1310 m_vset(10,V)
1320 /*
1330 /*          D r u m s   S e t
1340 /*
1350 /*          [ Kick ]
1360 B="Y2,23"
1370 B1=B+"R&"
1380 B2=B1+"R&"
1390 B3=B2+"R8&"
1400 /*          [ Snare ]
1410 S="Y2,15"
1420 S1=S+"R16&"
1430 S2=S+"R8&"
1440 S3=S+"R4&"
1450 /*          [ Tom Tom ]
1460 H="Y2,30"
1470 /*          [ Rest ]
1480 R1="R16&"
1490 R2="R8&"
1500 R3="R4&"
1510 /*          [ etc... ]
1520 CL="Y2,8"
1530 C="Y2,65"
1540 O="Y2,66"
1550 /*
1560 /*          M u s i c   D a t a
1570 /*
1580 /*
1590 /*            Synth 1
1600 /*
1610  X(0)="@1 V13 Y48,20 Q8 O4"
1620  X(1)="G+1& G+1 @3 L8 V13 Y48,20 O5"
1630  X(2)="D+4.C+2& C+1 >G+16<C+16 @L1"
1640  X(3)="Y48,0D   Y48,16D  Y48,32D   Y48,48D"
1650  X(4)="Y48,4D   Y48,80D  Y48,96D  Y48,112D"
1660  X(5)="Y48,128D Y48,144D Y48,160D Y48,176D"
1670  X(6)="D16&D4 Y48,0  L8 C+1& C+2>G+16<C+16
1680  X(7)="C+4.>B2& B1 E16B16 <@L1"
1690  X(8)="Y48,0C   Y48,16C  Y48,32C  Y48,48C"
1700  X(9)="Y48,64C  Y48,80C  Y48,96C  Y48,112C"
1710 X(10)="Y48,128C Y48,144C Y48,160C Y48,176C"
1720 X(11)="C16&C4> Y48,20 L8 B2& B1 R8"
1730 X(12)="|:2 B4.B-4.<D-1>A-4 F+1& F+1 :|L16 O3 Q6"
1740 /*
1750 X(13)="@4 V11 P3   |:2 G+D+A+D+ BBRR G+D+A+BR4 :|"
1760 X(14)="|:2 <C+>A+BG+ A+A+R <C+> A+BG+A+R4 :|"
1770 X(15)=X(13)
1780 X(16)=X(14)
1790 X(17)="|:2 BG+A+F+ G+ERBG+A+F+ER4 :|"
1800 X(18)="|:2 A+GG+F  GD+RA+ GG+FD+R4  :|"
1810 X(19)=X(17)
1820 X(20)=X(18)
1830 /*
1840 X(21)="V11 Q8 |:2 @8 @L1    O1 R4D+4G+4<"
1850 X(22)="Y48,20F  Y48,41F  Y48,62F  Y48,83F"
1860 X(23)="Y48,104F Y48,125F Y48,146F Y48,167F"
1870 X(24)="Y48,188F Y48,209F Y48,230F Y48,251F"
1880 X(25)="F8.       Y48,20 D+1&D+1&D+1 :| L16"
1890 /*
1900 X(26)="O2 V12 P3 F1&F2.D+FD+F F+1& F+4A+4G+4F+4"
1910 X(27)="G+1& G+4C+4F+4G+4  A1<C+1"
1920 X(28)="V12 |:3 D+1& :|D+2."
1930 /*
1940 for I=0 to 28 : m_trk(1,X(I)) : next
1950 /*
1960 /*          Synth 2
1970 /*
1980  X(0)="@1 V13 Y49,24 Q8 O4"
1990  X(1)="B1 & B1  @3 L8 V10 Y49,32 O5 R16"
2000  X(3)="Y49,48D  Y49,64D  Y49,80D  Y49,96D"
2010  X(4)="Y49,112D Y49,128D Y49,144D Y49,160D"
2020  X(5)="Y49,176D Y49,192D Y49,208D Y49,224D"
2030  X(6)="D16&D4   Y49,32 L8 C+1& C+2>G+16<C+16"
2040  X(8)="Y49,48C  Y49,64C  Y49,80C  Y49,96C"
2050  X(9)="Y49,112C Y49,128C Y49,144C Y49,160C"
2060 X(10)="Y49,176C Y49,192C Y49,208C Y49,224C"
2070 X(11)="R16&C4>  Y49,32 L8 B2& B1 R8"
2080 /*
2090 X(13)="@4 V7  P1 R |:2 G+D+A+D+ BBRR G+D+A+BR4 :|"
2100 X(15)=X(13)
2110 /*
2120 X(21)="V9  Q8 |:2 @8 @L1    O1 R4D+4G+4<"
2130 X(22)="Y49,40F  Y49,61F  Y49,82F  Y49,103F"
2140 X(23)="Y49,124F Y49,145F Y49,166F Y49,187F"
2150 X(24)="Y49,208F Y49,228F Y49,238F Y49,248F"
2160 X(25)="F8. Y49,40 D+1&D+1& |1 D+1 :|D+2.. L16 P3"
2170 /*
2180 X(26)="O1 V12 P3 F1&F2.D+FD+F F+1& F+4A+4G+4F+4"
2190 X(27)="G+1& G+4C+4F+4G+4 A1<C+1 V13"
2200 X(28)="|:3 D+1& :|D+2."
2210 /*
2220 for I=0 to 28 : m_trk(2,X(I)) : next
2230 /*
2240 /*          Synth 3
2250 /*
2260  X(0)="@1 V13 Y50,28 Q8 O5"
2270  X(1)="E1 & E1  @3 L8 V9  Y50,48 O5 R8"
2280  X(3)="Y50,40D  Y50,56D  Y50,72D  Y50,88D"
2290  X(4)="Y50,104D Y50,120D Y50,136D Y50,152D"
2300  X(5)="Y50,168D Y50,184D Y50,200D Y50,216D"
2310  X(6)="D16&D4 Y50,40 L8 C+1& C+2>G+16<C+16
2320  X(8)="Y50,48C  Y50,64C  Y50,80C  Y50,106C"
2330  X(9)="Y50,110C Y50,118C Y50,126C Y50,134C"
2340 X(10)="Y50,176C Y50,192C Y50,208C Y50,224C"
2350 X(11)="R16&C4> Y50,52 L8 B2& B1 R8"
2360 /*
2370 X(13)="@4 V7  P2   |:2 G+D+A+D+ BBRR G+D+A+BR4 :|"
2380 X(15)=X(13)
2390 /*
2400 X(21)="    Q8 |:2 @8 @L1 P1 O1 R4D+4G+4<"
2410 X(22)="Y50,48F  Y50,69F  Y50,90F  Y50,111F"
2420 X(23)="Y50,132F Y50,153F Y50,174F Y50,195F"
2430 X(24)="Y50,216F Y50,226F Y50,236F Y50,254F"
2440 X(25)="F8. Y50,52 D+1&D+1&D+1 :| L16"
2450 /*
2460 X(26)="O2 V10 F1&F2.D+FD+F F+1& F+4A+4G+4F+4"
2470 X(27)="G+1& G+4C+4F+4G+4 A1<C+1"
2480 X(28)="V11 |:3 D+1& :|D+2"
2490 /*
2500 for I=0 to 28 : m_trk(3,X(I)) : next
2510 /*
2520 /*          Synth 4
2530 /*
2540  X(0)="@1 V13 Y55,32 Q8 O5"
2550  X(1)="G+1&G+1 @2 Y51,20 O5 V12 L8 Q6"
2560  X(2)="D+C+>GEC+EG+<C+ |:2 EC+>GEC+EG+<C+ :|"
2570  X(3)="EC+>GEC+EA<C+> |:4 <D+>BA+G+D+G+A+B :|"
2580  X(4)="|:8 <C+>A+G+F+C+F+G+A+ :|<"
2590 /*
2600  X(5)="V11 L16 @2 o4"
2610  X(6)="|:4 G+<D+D+>G+ <D+D+>G+<D+> :|"
2620  X(7)="|:4 A+<F+F+>A+ <F+F+>A+<F+> :|"
2630  X(8)=X(6)
2640  X(9)=X(7)
2650 /*
2660 X(10)="V10|:4 EBBE BBEB :||:4 D+A+A+D+ A+A+D+A+ :|"
2670 X(11)=X(10)
2680 X(12)="@8 V5 O1 L1 Q8"
2690 X(13)="|:2 G+& G+2..&G+24& y2,36G+24& Y2,37g+24& G+&G+ :|"
2700 /*
2710 X(14)="L16 v11 o0 A+1& A+1 B1& B1<"
2720 X(15)="C+1& C+1 D1 E1"
2730 X(16)="V11 |:2 Y2,37 G+1& |1  G+1 :|G+2."
2740 /*                              (注)トラック8です。
2750 for I=0 to 16 : m_trk(8,X(I)) : next
2760 /*
2770 /*          Synth 5
2780 /*
2790  X(0)="@1 V13 Y52,36 Q8 O5"
2800  X(1)="B1 &B1  @2 Y52,36 O5 V7  L8 Q6 R16"
2810  X(5)="|:4 R1 :| @8 Y52,40 V12 O2 R16 Q8"
2820  X(6)="|:3 d+1& :|D+2... D+32F+32"
2830  X(7)="G+1&G+2...<D+32C+32> A+1&A+2...D+32F+32"
2840  X(8)="G+1&G+2...<D+32C+32> A+2...B16<C+2D+4. O1"
```

```
2850  X(9)="V8 R8 |:2 @8 P2 @L1 O1 R4D+4G+4<"
2860 X(10)="Y52,40F  Y52,81F  Y52,102F Y52,123F"
2870 X(11)="Y52,124F Y52,145F Y52,166F Y52,187F"
2880 X(12)="Y52,208F Y52,229F Y52,250F Y52,254F"
2890 X(13)="F8. Y52,48 D+1&D+1&D+1 :| O1 L16  P2"
2900 /*
2910 X(14)="A+1& A+1 B1& B1<"
2920 X(15)="C+1& C+1 D1 E1"
2930 X(16)="V12 |:3 G+1& :|G+2"
2940 /*
2950 for I=0 to 16 : m_trk(5,X(I)) : next
2960 /*
2970 /*          ●  Synth 6  ●
2980 /*
2990  X(0)="@1 V10 Y53,48 Q8 O5 R8 P1"
3000  X(1)="B1 &B2..@5 Y53,20 O3 V11 L32 R1 P1"
3010  X(2)="Y53,252D&  Y53,220D&  Y53,188D&  Y53,156D&"
3020  X(3)="Y53,124D&  Y53,92 D&  Y53,60 D&  Y53,28 D&"
3030  X(4)="Y53,252C+& Y53,220C+& Y53,188C+& Y53,156C+&"
3040  X(5)="Y53,124C+& Y53,92 C+& Y53,60 C+& Y53,28 C+&"
3050  X(6)="Y53,252C&  Y53,220C&  Y53,188C&  Y53,156C&"
3060  X(7)="Y53,124C&  Y53,92 C&  Y53,60 C&  Y53,28 C& >"
3070  X(8)="Y53,252B&  Y53,220B&  Y53,188B&  Y53,156B&"
3080  X(9)="Y53,124B&  Y53,92 B&  Y53,60 B&  Y53,28 B&"
3090 X(10)="Y53,252A+& Y53,220A+& Y53,188A+& Y53,156A+&"
3100 X(11)="Y53,124A+& Y53,92 A+& Y53,60 A+& Y53,28 A+&"
3110 X(12)="Y53,20 R2. R1"
3120 X(13)="@9 V11 O0 L2 G+A+B<C+D+EF+G+"
3130 X(14)="@10 V8 O1 L8 |:2  R1 F+A+B<C+ D+E4<E1&E1r > :|"
3140 X(15)="|:4 R1 :| @8 V15 O2"
3150 X(16)="|:3 d+1& :|D+2... D+32F+32"
3160 X(17)="G+1&G+2...<D+32C+32> A+1&A+2...D+32F+32"
3170 X(18)="G+1&G+2...<D+32C+32> A+2...B16<C+2D+2 O1"
3180 /*
3190 X(19)="V11 |:2 @8 @L1 O2 R4D+4G+4<"
3200 X(20)="Y53,20F  Y53,41F  Y53,62F  Y53,83F"
3210 X(21)="Y53,104F Y53,125F Y53,146F Y53,167F"
3220 X(22)="Y53,188F Y53,209F Y53,230F Y53,251F"
3230 X(23)="F8. Y53,20 D+1&D+1&D+1 :| O1 L16 V11"
3240 /*
3250 X(24)="A+1& A+1 B1& B1<"
3260 X(25)="C+1& C+1 D1 E1"
3270 X(26)="V12 |:3 G+1& :|G+2."
3280 /*
3290 for I=0 to 26 : m_trk(6,X(I)) : next
3300 /*
3310 /*          ♨  Synth 7 / Bass 1  ♨
3320 /*
3330  X(0)="@1 V10 Y54,48 Q8 O5 R8 P2"
3340  X(1)="B1 &B2..@5 Y54,40 O3 V11 L32 R1 P2 @L1R L32"
3350  X(2)="Y54,236D&  Y54,204D&  Y54,172D&  Y54,140D&"
3360  X(3)="Y54,108D&  Y54,76 D&  Y54,44 D&  Y54,12 D&"
3370  X(4)="Y54,236C+& Y54,204C+& Y54,172C+& Y54,140C+&"
3380  X(5)="Y54,108C+& Y54,76 C+& Y54,44 C+& Y54,12 C+&"
3390  X(6)="Y54,236C&  Y54,204C&  Y54,172C&  Y54,140C&"
3400  X(7)="Y54,108C&  Y54,76 C&  Y54,44 C&  Y54,12 C&>"
3410  X(8)="Y54,236B&  Y54,204B&  Y54,172B&  Y54,140B&"
3420  X(9)="Y54,108B&  Y54,76 B&  Y54,44 B&  Y54,12 B&"
3430 X(10)="Y54,236A+& Y54,204A+& Y54,172A+& Y54,104A+&"
3440 X(11)="Y54,108A+& Y54,76 A+& Y54,44 A+& Y54,12 A+&"
3450 X(12)="Y54,40 R2 R8...@L1RR R1R8"
3460 X(13)="@9 V8  O0 L2 G+A+B<C+D+EF+G+4."
3470 X(14)="@10 V6 O1 L8 |:2 RR1 F+A+B<C+ D+E4<E1&E2..r> :|"
3480 X(15)="@7  O2 V14 Y54,20"
3490 X(16)="|:3 G+1&G+2...D+32F+32 :|"
3500 X(17)="|:2 G+1&G+2...D+32E32  :|"
3510 X(18)="|:2 G+1&G+2...D+32 |1 G32 :|E32 G+1&G+1"
3520 X(19)="Y52,40 V12 @6 |:9 G+1&G+1& :| G+1&G+2R8"
3530 /*
3540 for I=0 to 19 : m_trk(7,X(I)) : next
3550 /*
3560 /*          ■  Drums & Bass2  ■
3570 /*
3580  X(0)="Y3,3 L16 Y51,20 Q8 @V0"
3590  X(1)="R1R1"
3600  X(2)="|:2 "+B3+R2+B1+B3+R1+CL+R3
3610  X(3)=B3+R2+B1+B3+R1+R3
3620  X(4)=B3+R2+B1+B3+R1+H+R3
3630  X(5)=B3+R2+B1+B3+R1+R3+" :|"
3640  X(6)="|:3 "+B3+R2+B1+B3+R1+CL+R3+" :|"
3650  X(7)=B3+R2+B1+B3+R1+CL+R2+CL+R2
3660  X(8)=X(6)
3670  X(9)=X(7)
3680 /*
3690 ZA="V14 @6 O2"
3700 X(10)=ZA+"|:7 "+B+"G+4&"+S2+B1+B3+R1+S3+":|"
3710 X(11)=B3+S2+B1+B3+R1+S2+S1+S1
3720 X(12)="|:3 "+B3+S2+B1+B3+R1+S3+":|"
3730 X(13)=B3+S2+B1+B3+R1+S1+C+R1+O+R2
3740 X(14)="|:4 "+B3+S2+B1+B3+R1+S3+":|"
3750 /*
3760 ZB=B3+S2+B1+B2+B2+R1+S3
3770 X(15)=ZB+ZB+R3+S2+B1+B2+B2+R1+S3
3780 X(16)=B3+S2+B1+B2+B2+R1+"|:4 "+S1+" :|"
3790 X(17)=X(15)
3800 X(18)=B3+S2+B1+B2+B2+R1+"|:2 "+S2+" :|"
3810 X(19)="|:3"+ZB+":|"
3820 X(20)=B3+S2+B1+B2+B2+R1+S2+S1+S1
3830 X(21)="|:4 "+B3+S2+B1+B2+B2+R1+S3+":|"
3840 X(22)="|:2 R4&"+S2+B1+B2+B2+R1+S3
3850 X(23)="|1 "+B3+S2+B1+B2+B2+R1+S3
3860 X(24)=":| "+B3+S2+B1+B2+B2
3870 X(25)="R16 @V0"+CL+R2+CL+R1+CL+R1
3880 /*                                  (注)トラック4です。
3890 for I=0 to 25 : m_trk(4,X(I)) : next
3900 /*
3910 m_play()
```

X68000用

キューピー3分クッキングのテーマ

小宮山 博志 Komiyama Hiroshi

今月のイロモノ

キューピー3分クッキングのオープニングテーマをお送りします。この曲は誰もが聞いたことあるのではないでしょうか。この作品ではオーケストラヒットを使用した，豪華絢爛フルコーラスバージョンに仕上がっています。作者の小宮山君によると，正しい遊び方が存在しています。

アンプ	1つ
ラジカセ	少々
イコライザ	カップ1杯

を用意してください。

さて，準備ができましたら，イコライザの100Hz以下が100%，500～2kHzが50～60%，10kHz以上を80～90%にします。お好みに合わせて，バスブーストを入れるとか，サラウンドにするなど，彩りを添えてくださいね。

聴くときは，左右のスピーカーとの正三角形の頂点より多少遠くの位置でなるべく大音量で楽しみましょう。「てけてんてんてんてんてん」と口ずさむと一層楽しく聴くことができます。

さて，明日はペヤングソース焼きソバの作り方をお送りしましょう（しないって)。

リスト3　キューピー3分クッキング

```
 10 /*
 20 /*キューピー・3分クッキングのBGM
 30 /*
 40 char v(4,10)
 50 v={44,15, 0, 0, 0, 0, 0, 0, 0, 3, 0,
 60     31, 0, 0, 0, 0,33, 0, 4, 3, 0, 0,
 70     31,11, 0, 6, 6, 0, 1, 2, 3, 0, 0,
 80     31, 0, 0, 0, 0,31, 0, 6, 7, 0, 0,
 90     31,11, 0, 6, 6, 0, 1, 2, 7, 0, 0}
100 m_vset(70,v)
110 m_init():for i=1 to 8:m_alloc(i,10000):m_assign(i,i):next
120 /*
130 str a(20)[256]
140 /*
150 a(0)="t160@70o4l16v14y3,3":a(1)="r8gggrgrererfr8. r8gggrgr
c+rc+rdr8.
160 a(2)="t168r8gg grf+r frer d+rdr c+rcr t164>br8.<gr8.>br8.<
grt160gg
170 a(3)="er8.gr8.gr8.r8gg fr8.gr8.gr8.r8gg er8.gr8.br8.ar8. g
rf+rfrerdrr4gg
180 a(4)="er8.gr8.gr8.r8gg dr8.br8.br8.r8bb arrrarrr<drrrdrrr
>grrr>gg+aa+brgrargr
190 a(5)="errrgrrrergrrrgr frrrgrrrfrgrrrgr errrgrrrbrrrarrr
200 a(6)="grf+rfrerdrr4gg errrgrrrgrrrrrgg frrrarrrarrr<crrr
210 a(7)="errrr4drrrr4 cr>f+gg+aa+b<crrrc4
220 for i=0 to 7:m_trk(1,a(i)):next
230 /*
240 a(0)="@70o4l16v11p2r16y54,10":for i=0 to 7:m_trk(7,a(i)):n
ext
250 /*
260 a(0)="@70o4l16v11p1r16.y55,30":for i=0 to 7:m_trk(8,a(i)):
next
270 /*
280 a(0)="@70o4l16v13y49,40p2 r2y2,45c+ry2,45c+ry2,46drrr r2>y
2,42b-ry2,42b-ry2,43brrr
290 a(1)="v12<r4r12grf+rfrerd+rdrc+rcr>brrr<grrr>brrr<grr24 p3
v12o3g2e2 b2g2 <c2>e2 b2g2
300 a(2)="<c2>g2 b2g2 a2f+2 grrrr2. o4c2>g2 b2g2 <c2>g2 b2g2 <
c2>g2 a2f2 g2b2 <c2r4e4
310 for i=0 to 2:m_trk(2,a(i)):next
320 /*
330 a(0)="@70o4l16v10y50,0p1 r1r1 r4r8.grf+rfrerd+rdrc+rcr>brr
r<grrr>brrr<g
340 a(1)="v15o2q3p3l4c>g<c>g dgdg <c>g<c>g dgdg <c>g<c>g dgdg
adad gr2.
350 a(2)="o2c>g<cc >gdgg <c>g<cc >gdgg <c>g<cc >fcff eghg <c>g
<cq8c
360 for i=0 to 2:m_trk(3,a(i)):next
370 /*
380 a(0)="@70o3v1l18q6y51,0":a(1)="r1r1r1r2..y2,48ry2,48ry2,48
r
390 a(2)="|:rererererfrfrfrf:|rererere rgrgrgrg rf+rf+rf+rf+ y
```

```
2,48grr4.y2,48rry2,48r
  400 a(3)="|:3rerererefrfrfrf:| y2,45rerey2,43rfrfy2,41rererry
2,41r
  410 for i=0 to 3:m_trk(4,a(i)):next
  420 /*
  430 a(0)="@70o3v1118q6p3y51,0"
  440 a(2)="|:rcrcrcrc>rbrbrbrb<:|rcrcrcrc|:rdrdrdrd:| drr2.
  450 a(3)="|:rcrcrcrc>rbrbrbrb<:||:rcrcrcrc:| rcrc>rbrb< rcrcr
```

```
460 for i=0 to 3:m_trk(5,a(i)):next
470 /*
480 a(0)="@70o2v1118q6p3y51,0
490 a(2)="|:5rgrgrgrg:|rbrbrbrbrararara brr2.
500 a(3)="|:5rgrgrgrg:|rararara rgrgrgrg rgrgr
510 for i=0 to 3:m_trk(6,a(i)):next
520 m_play()
```

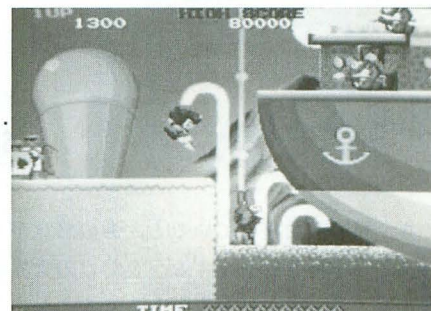
X1/turbo用

Marvel Land

©NAMCO

牧野 守弘 Makino Morihiro

音楽性が高いナムコミュージック

ふと気がついたのですが，ナムコのゲームミュージックがやたらに載ってるんじゃないの？　このページって。特に意味はないんですけど。最近のゲームミュージックでは，やはりナムコがいちばん出来がいいみたいですね。聴いてても楽しいし。このMarvel Landは西川さんが基板を持っていますので（要するにシステム2を持っている），彼の家でコインを気にせずに遊んだものです。さて，作品のほうですが，もともとが短い曲であったこともあり，おまけのサブルーチンを含めてみても，短めになっています。入力しやすいのは嬉しいことですね。音色のほうがイマイチ似ていないといわれてしまいそうですが，ノリは完全にMarvel Landになっていますので，よしとしましょう。おまけのサブルーチンは30行から140行までで，音色のリストを出力するサブルーチンと，各音色のデータをダンプするサブルーチンです。曲を演奏するときには必要ありませんが，この手のサブルーチンはあると便利ですので，ぜひ入力しましょう。(S.K.)

リスト4　Marvel Land

```
10 ' Marvel Land
20 GOTO 150      'It is no nessesary to input 30-140.
30 LABEL"TONE LIST
40 FOR N=0 TO 39
50   PRINTUSING"   ##:";N+1;
60   PRINT#0 MEM$(&HB000+N*10,10);
70 NEXT
80 RETURN
90 LABEL"OPMDATA"
100 INPUT"Tone No.=";TN:T=TN-1
110 PRINT#0 "DATA";TN;",";MEM$(&HB000+10*T,10)
120 FOR N=0 TO 1:PRINT"DATA ";:FOR M=0 TO 17
130   PRINT RIGHT$("0"+HEX$(PEEK(&HB190+T*36+M+N*18)),2)+" ";
140 NEXT:PRINT:NEXT:RETURN
150 '———————————————————————————— O P M    D A T A  -
160 DATA 12 ,Guiter  *5
170 DATA FD 00 45 31 43 71 28 00 2B 00 1F 1F 1F 1F 08 0A 0A 0A
180 DATA 08 08 08 08 55 34 34 34 00 00 00 00 00 00 80 00 00 00
190 DATA 13 ,Trumpet *1
200 DATA D2 00 01 11 42 51 11 1B 1B 00 11 10 0F 0D 0E 00 00 08
210 DATA 00 00 00 00 F0 00 00 0A 00 00 00 00 00 00 80 00 00 00
220 DATA 14 ,Guiter  *4
230 DATA ED 00 33 40 42 71 1A 38 00 00 1F 1F 1F 9F 00 06 06 06
240 DATA 00 1F 1F 1F 00 F4 F4 F4 00 00 00 00 00 00 80 00 00 00
250 DATA 17 ,Violin  *2
260 DATA C3 00 45 71 43 31 2F 26 20 00 1F 12 12 4F 1F 1F 03 08
270 DATA 00 00 03 05 00 00 F0 25 00 00 00 00 00 00 80 00 00 00
280 DATA 32 ,namco Drum
290 DATA FE 00 4F 41 41 41 0D 03 00 00 1F 1F 1F 1F 00 0F 07 07
300 DATA C0 14 1F 1F 0F 6F FF FF 00 00 00 00 00 00 80 00 00 00
310 DATA 35 ,Guiter  *3
320 DATA C0 00 42 10 53 41 1B 1E 1E 00 9F 9F 9F 9F 00 03 03 07
330 DATA 00 00 03 03 00 F0 F0 A6 00 00 00 00 00 00 80 00 00 00
340 DATA 36 ,Base   *S1
350 DATA ED 73 30 71 31 10 1B 70 43 43 9F 5E 5F DF 05 05 09 03
360 DATA 05 03 04 02 97 45 A6 47 00 00 00 00 00 00 80 00 00 00
370 DATA 39 ,Snare Drum
380 DATA FD 00 0C 01 01 01 00 61 61 61 1F 5C 5E 9C 00 91 11 8F
390 DATA C0 00 00 00 01 F8 D9 F7 00 00 00 00 00 00 80 00 00 00
400 DATA 0,0,0,0
410 '------------------------------------ SET TONE DATA    -
420   READ T,TN$,A$,B$:IF T=0 THEN 460
430   MEM$(&HB000+(T-1)*10,10)=TN$
440   MEM$(&HB190+(T-1)*36,36)=HEXCHR$(A$+B$)
450 GOTO 420
460 LABEL"HS"
470 TEMPO0
480 PLAY"T121"
490 '———————————————————————————— Guiter 1 / Trumpet -
500 PLAY"R128[I14 L16 O5 V13 ";
510 PLAY"GECAFCG8E8C4.  GECAFCG8E2  GECAFCG8E8C4.  <G>F<G>F<G>FE
8D2";
520 PLAYSTRING$(3,"D<BF>FD<F>E8D2")+"<G>G<G>G<G>GF8E2";
530 PLAY"GECAFCG8E8C4.  GECAFCG8E2  GECAFCG8E8C4.  <G>F<G>F<G>FE
8D2";
540 PLAYSTRING$(3,"D<BF>FD<F>E8D2")+">C<GAB>CCC4.<CDE8";
550 PLAY"V16L8I13O5A2&AG+A>C   &C2&CC<BA G2&GEG>C  &C1";
560 PLAY"<A2&AG+A>C  &C1  D2&D<AB>C< B1 ]";
570 '———————————————————————————— Violin          -
580 PLAY":I17 L8 O4 V16 R128[";:V$="EGFAGEFD"
590 PLAY V$+V$+V$+"D<BGDFGB>C";
600 PLAY STRING$(2,"DFEGFD<B>C")+"DFEGFD<AB  >CECGCECG";
610 PLAY V$+V$+V$+"D<BGDFGB>C";
620 PLAY STRING$(2,"DFEGFD<B>C")+"DFEGFD<AB  >CECGCECG";
630 PLAY"AF<B>CAF<B>C  AF<B>CAF<B>A  GEFAGEFA GEFAGEFA";
640 PLAY"AF<B>CAF<B>C  AF<B>CF<A-B>D D<AF+D>D<AFD  B>D<BGB>D<BG>
]";
650 '———————————————————————————— Guiter 3-1        -
660 PLAY":I12 L8 O5 V12 H7 R128[";
670 PLAY"EEF4EC4E  &EEEE2R  EEF4E4C4  D1";
680 PLAY"DDE4D4.D  &DDEED2  DDE4D4F4  E1";
690 PLAY"EEF4EC4E  &EEEE2R  EEF4E4C4  D1";
700 PLAY"DDE4D4.D  &DDEED2  DDE4D4F4  E1";
710 PLAY"R1 R1 R1 R1 ";
720 PLAY"R1 R1 R1 R1]";
730 '———————————————————————————— Guiter 3-2        -
740 PLAY":I12 L8 O5 V12 H7 R128[";
750 PLAY"CCC4C<G4>C  &CCCC2R  CCC4C4<G4  B1";
760 PLAY"BBB4B4.B    &BBBBB2  BBB4B4>D4  C1";
770 PLAY"CCC4C<G4>C  &CCCC2R  CCC4C4<G4  B1";
780 PLAY"BBB4B4.B    &BBBBB2  BBB4B4>D4  C1";
790 PLAY"R1 R1 R1 R1 ";
800 PLAY"R1 R1 R1 R1]";
810 '———————————————————————————— Guiter 2          -
820 PLAY":I35 L16 O5 [V14";
830 PLAYSTRING$(6,"C8E8FG>C8<")+STRING$(8,"<G8B8>CDG8");
840 PLAYSTRING$(8,"C8E8FG>C8<")+STRING$(8,"<G8B8>CDG8");
850 PLAYSTRING$(2,"C8E8FG>C8<")+"V15";
860 PLAYSTRING$(4,"<F8A8B>CF8")+STRING$(4,"C8E8FG>C8<");
870 PLAYSTRING$(3,"<F8A8B>CF8")+"<F8A-8B>CF8"+STRING$(2,"<F+8A8B
>CF8");
880 PLAY"<G8>V16B>C<<G>>C8.<G>D<B>D<G>D<B>D<]";
890 '———————————————————————————— namco Drum        -
900 PLAY":I32 L8 O3 V15 S4,1,0,60 =3[";
910 PLAYSTRING$(23,"RRERRRER")+"RRERRREE16E16]";
920 '———————————————————————————— Base              -
930 PLAY":I36 L8 O3 V15[";
940 PLAYSTRING$(3,"CCG>C<CCG>C<  ")+STRING$(4,"<GGB>D<GGB>D")+"C
CG>C<CCG>C<";
950 PLAYSTRING$(3,"CCG>C<CCG>C<  ")+STRING$(4,"<GGB>D<GGB>D")+"C
CG>C<CCG>C<";
960 PLAYSTRING$(2,"<FF>C<F>FF>C<F")+STRING$(2,"CCG>C<CCG>C<");
970 PLAY"<FF>C<FFF>C<F  FF>C<FFFA->C  <F+F+>DF+<F+F+>DF  <GG>DG<
GG>DG]";
980 '———————————————————————————— Snare             -
990 PLAY":I39 L4 O3 V15";
1000 PLAY"[E]";                                    '[Tenu:]
1010 '——————————————————————————— Hi-Hat            -
1020 PLAY":L8 O8 S4,1,14,0 =3 V15 Y7,49 Y6,0[";
1030 PLAY STRING$(190,"E")+"RR]";
1040 '——————————————————————————— Guiter 4-1        -
1050 PLAY":L8 O4 V15 S0,0,0,0 =1 ^1 K0 [";
1060 PLAY STRING$(2,STRING$(12,"RG")+STRING$(16,"RF")+STRING$(4,
"RG"));
1070 PLAY STRING$(8,"RF")+STRING$(8,"RG")+STRING$(12,"RF")+"RGRG
RFRF]";
1080 '——————————————————————————— Guiter 4-2        -
1090 PLAY":L8 O4 V15 S0,0,0,0 =1 ^1 K0 [";
1100 PLAY STRING$(2,STRING$(12,"RE")+STRING$(16,"RD")+STRING$( 4
,"RE"));
1110 PLAY STRING$(8,"RC")+STRING$(8,"RE")+STRING$( 8,"RC")+STRIN
G$(8,"RD")+"]"
1120  '=-=-=-=-=-=-=-=-=-=-  References  -=-=-=-=-=-=-=-=-=-=-'
1130  '                                                       '
1140  'Knight Arms 3D-Stage by Nishikawa Zenji (Oh!X '90年06月 )'
1150  '                         PSG Hi-Hat                    '
1160  '                         PSG Parametors                '
1170  '                                                       '
1180  'Marvel Land  / namco   (SCITRON)                       '
1190  '                         Score                         '
1200  '                                                       '
1210  '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-'
```

清水和人流
プログラミング道場

ああ,ピアニスト

〔その2〕

Shimizu Kazuto

演歌プログラミングで日本人の心を歌った清水和人氏が今月はショパンになる。さあ君も暇プロ詩人となってキーボードを白鍵と黒鍵に塗り分け（?），珠玉の名演奏を末永く後世に残そうではないか。

夜，机の前に座っていると，あの感覚がヒタヒタと押し寄せてくる。夜だけの世界がボンヤリとした頭を支配してくる。かすかに聞こえる呼び声，それは遠くに，しかしハッキリとこう言っている。「さぁ，BASICを起動するのだ」と。

そうだ，いまこそBASICを起動しなければならない。そこに広がる無限大の世界を見るために。そこを自分の世界に変えていくあの感覚を味わうために。人類の生み出した輝かしい新文明，ソフトウェアの世界を切り開くために。

キーボードをキーボードに

いつものように口をポカンとあけ，暇プログラミング商店街を散歩していると，指がふとタイピング感覚からショパンに変わってしまった。そういえばピアノもしばらく弾いていないなぁ。そう思うといままでやっていたプログラムなど忘れて，指づかいはすっかりピアノ。画面もメチャクチャになってしまった。

これはいかん，と思ってちょちょいのちょいと作ったのがリスト1のプログラムである。X68000のキーボードを演奏用のキーボードに変えてしまったのだ。その短さはBASICマニュアルに載せられるほど小さい。20，30，60行の文字列の初期化がいやらしいけど，ビールを飲みながら十分打ち込める程度である。

X-BASICにはFM音源のコントロールをする関数が用意されているのだから，BASICマニュアルに載っているMML（音楽を記述するための言語）とその例題さえ読めば，すぐに音楽演奏が可能である。

70行はメモリの確保，80行はメモリをチャンネルごとに割り当てている。ここでは8チャンネルのうち6チャンネルを使っている。あとは，140～160行でMMLを書き込んではFM音源を鳴らしている。

20行のba$は，キーボードのどれが押されたかを判断するために使っている（キーボードの配置は鳴らしてみて覚えてね）。それが30行と60行で定義したca$の何番目というふうに対応していて，お望みの音が出るって寸法である。

噂によると，このままだとピアノだが，150行の@1を@15に変えるオルガンになるらしい。この数字と音色の関係はBASICマニュアルの51ページ（初代機種の場合は116ページ）に載っていること請け合いである。最近，小さなキーボードがもういやっていうほどいろいろ出ているけど，音色を変えて演奏するだけならX68000で十分ってわけだ。

そうそう，もうひとつ忘れていた。90行のinkey関数に引数0が与えられているが，マニュアルには載っていない。これは，どのキーも押さないでいると押されるまで待たずに次へ進むものだ。リアルタイムキー入力をするとき必ず使うので覚えておきたい。

●リスト1

```
   10 int f=0
   20 str ba$[46]="zxcvbnm,./qwertyuiop@[asdfghjkl;:1234567890-^
¥"
   30 str ca$(46)[3]={"3c","3d","3e","3f","3g","3a","3b","4c","4
d","4e","4f","4g","4a","4b","5c","5d","5e","5f","5g","5a","5b","
6c","2b","3c+","3d+","3e+","3f+","3g+","3a+","3b+","4c+","4d+","
4e+","4e+","4f+","4g+","4a+","4b+","5c+","5d+","5e+","5f+"}
   40 str e$[4]
   50 str  a$=""
   60 ca$(42)="5g+":ca$(43)="5a+":ca$(44)="5b+":ca$(45)="6c+"
   70 for i=1 to 6:m_alloc(i,2000):next
   80 for i=1 to 6:m_assign(i,i):next
   90 a$=inkey$(0)
  100 if a$="" then goto 90
  110 f=instr(1,ba$,a$)-1
  120 if f=-1 then goto 90
  130 e$="o"+ca$(f)
  140 m_init()
  150 m_trk(1,"@1"+e$)
  160 m_play()
  170 goto 90
```

X-BASICの特徴

ところで，前回は触れなかったが，X-BASICは従来のBASICとちょっと違うのである。これがX1などのBASICを使っていた人が，つい足を止めてしまうひとつの原因かもしれない。しかし，リスト1を見るとたいして変わらないじゃないかと思う人も多いだろう。そう，気にすることはないのである。ただし，一応気をつけることはある。たとえば変数の宣言である。これさえやっておけば，どうということはない。

あと，X-BASICの特徴を生かしたプログラムをするなら，gotoはできるだけ使わないほうがよい。repeat～untilやwhile～endwhile，if～then～elseを巧みに使い分けるのが重要である。なぜなら，それが構造化プログラミングってやつだからである。うっそー，本当はgotoなどの飛先の行番号がrenumでリナンバーしたときに変更されないからである。おかげさまで，最後のプログラムではrepeat～untilを連発してしまったのであった。

いや～,なんてったってgotoこそがプロ

グラムの根本ですよ。皆さんも自分だけのプログラムを作るときはgotoをふんだんに使っておいしいスパゲッティプログラムにしていきませんかぁ。だって，repeat～untilやwhile～endwhileなんかを使うと，頭も使わなくてはならないじゃない。1つや2つならいいけど，10組も出てくるともう私の頭は睡眠の準備を始めてしまうのだぁ。

しかもである。数値計算の大御所FORTRANでも，goto文はスリリングな命令である。これをif～then～elseやwhile～endwhileにすると，どこが終わりなのかわからないような長い分岐のときに困るではないの。gotoのほうがエディット作業も楽なのであーる。goto万歳！

録音だぁ

さて，リスト1のあとでなぜgotoを使うことを称賛したのか？　それは一重にリスト2の言い訳なのである。見よ，goto文が増えて訳がわからなくなってきた。

いや～，訳がわからなくてもいいじゃないですか。そりゃ一瞬戸惑いますけどねぇ。goto文ならボケーと順番に追いかけるだけで必ず氷解するってもんですよ。

さて，gotoを使うのはもうひとつ巨大な理由がある。それは，

「あとから少しずつ付け加えるプログラムに便利だから」

なのである。最初からプログラムを設計しておいてから作るのではなく，暇にまかせて徐々に機能を加えていくのだ。そうなったら，repeat～untilやwhile～endwhileはものすごく不便。これは経験則である。

しかし，ああ無情にもリスト2のように10番おきの行番号のなかに177行とか275行なんてぇのが入ってしまうのである。

まあ，それでもいいじゃん，という考え方もある。人が書いたプログラムが理解できるようになると変な行番号のついているところが密かな楽しみになるのである。

「フーム，ここはあとから付け足したな。なんでだろう。えーと，そうかこういう勘違いをしていたに違いない。ナルホド」

などと思いを馳せ，あたかもそのプログラマと知り合いになったかのような気持ちになれるのである。あとから行番号をリナンバーして，その恥部を隠してしまうなんて人間が小さいなぁ。

ところで，リスト2には録音の機能が備わった。これであなたの奏でる魅惑の旋律を地球滅亡のその日まで保存することさえできるのである。さっきまでは，いたずら小僧のピアノぶったたきにすぎなかった旋律が，再生できることで芸術と化したかもしれないのである。その大きな夢を肴に一杯やったって，今夜は満足のいく睡眠がとれそうではないか。

そうだ，偉大な芸術家になる方に言っておかなければならないことがあった。キーボードの配置は，下段“Z”から“/”へ向かってドレミファ……と上がっていき，ミまできたら一段飛ばして“Q”から右の“［”へ向かってファソラシ……ときて“ド”までの3オクターブである。“S”からの段と“2”からの段は黒鍵である。慣れればそう難しいことではないが，いやな人はba$を変えれば変更できる。

また，判断は英数字の小文字で行うので，CAPSキーなどははずしておかねばならない。音は最大100音まで録音でき，それが終わるかリターンキーを押すかすると再生する。これらの設定を変えるにはプログラムをちょいといじるしかない。人のプログラムをいじって変えるのはもっともよいプログラム勉強法のひとつである。

140～280行が録音する場所。290行以降が再生ルーチンである。リスト1と同様，150行，350行を変えれば音色も変わる。配列c$に入力文字列を，配列dにキーが押されたタイミングを保存しておき，再生部でそれを参照しながら再生している。190行は押したキーを表示しているところで，なくてもよい。370～400行は音の長さ分だけ空回しを行っている。

このプログラムの欠点のもっとも大きなところは180行である。前に押されたキーを続けて押すと音がならないようにしている。これはX68000のキーリピートに対す

●リスト2

```
  10 dim str  c$(100)
  20 dim int d(100)
  30 int j=1:int f=0
  40 str ba$[46]="zxcvbnm,./qwertyuiop@[asdfghjkl;:1234567890-^
¥"
  50 str ca$(46)[3]={"3c","3d","3e","3f","3g","3a","3b","4c","4
d","4e","4f","4g","4a","4b","5c","5d","5e","5f","5g","5a","5b","
6c","2b","3c+","3d+","3e+","3f+","3g+","3a+","3b+","4c+","4d+","
4e+","4e+","4f+","4g+","4a+","4b+","5c+","5d+","5e+","5f+"}
  60 str e$[4]
  70 str  b$=""
  80 int i=0
  85 int jmax=100
  90 str  a$=""
 100 ca$(42)="5g+":ca$(43)="5a+":ca$(44)="5b+":ca$(45)="6c+"
 110 m_init()
 120 for i=1 to 6:m_alloc(i,2000):next
 130 for i=1 to 6:m_assign(i,i):next
 140 /*a$=inkey$
 150 a$=inkey$(0)
 160 i=i+1
 170 /*if i mod 100 =0 then beep
 175 if a$="" then goto 140
 177 if a$=chr$(13) then jmax=j:goto 290
 180 if a$=b$ then goto 140
 190 print a$;
 200 c$(j)=a$:d(j)=i:b$=a$
 210 f=instr(1,ba$,a$)-1
 220 if f=-1 then goto 270
 230 e$="o"+ca$(f)
 240 m_init()
 250 m_trk(1,"@1"+e$)
 260 m_play()
 270 j=j+1
 275 if j=jmax then goto 290
 280 goto 140
 290 for j=1 to jmax-1
 300 a$=c$(j)
 310 f=instr(1,ba$,a$)-1
 320 if f=-1 then goto 380
 330 e$="o"+ca$(f)
 340 m_init()
 350 m_trk(1,"@1"+e$)
 360 m_play()
 370 for i=1 to d(j+1)-d(j)
 375 /*print d(j+1)-d(j),d(j+1),d(j)
 380 for k=1 to 10
 385 /*b$=inkey$(0)
 390 next
 400 next
 410 next
 420 j=1:jmax=100:goto 140
```

る対策である。あるキーを押したままにすると，この180行がなければ，その音が断続的に繰り返されてしまう。そこでこの行が付け加えられたものである。そのため，同じ音が2回続いたとき，2個目が無視されてしまうのであった。この欠点は今回のプログラムでいちばん悩んだところである。

いずれにしても，巷で売られているメロディ記憶機能付き電子キーボードの機能はこの程度であるから，いかにX68000が素晴らしいかわかってこようというものである。メモリの量だってこっちのほうが断然多いし，欠点といえば，持ち運びにはちょっと大きすぎるということくらいである。

さて，こうなってくるとなんとなくさらにその上を目指したくなってくる。しかし，慌ててはいけない。ここでじっくり一晩休むのがコツである。“急いては事を為損じる”というやつで，次への秘策をよく考えたほうがよい。なんでも一気にやってしまおうとするとどこかで無理がくるのである。だからこその暇プロなのだ。ちょっとずつ作って大きくしていく雪ダルマ方式がよい。コツコツ増やしていく精神はいかにも日本的ではないか。そしてまた，ちょっとずつ増やしていくためにこそgoto文があるのである（くどいかな）。

●リスト3

```
   10 dim str c$(100,8)
   20 dim int d(100,8)
   30 int j=1,f=0,i=0,io=0,k=0,jmax=100,l=1
   40 str ba$[47]="zxcvbnm,./qwertyuiop@[asdfghjkl;:1234567890-^
¥ "
   50 str ca$(47)[3]={"3c","3d","3e","3f","3g","3a","3b","4c","4
d","4e","4f","4g","4a","4b","5c","5d","5e","5f","5g","5a","5b","
6c","2b","3c+","3d+","3e+","3f+","3g+","3a+","3b+","4c+","4d+","
4e+","4e+","4f+","4g+","4a+","4b+","5c+","5d+","5e+","5f+"}
   60 str cb$(47)[3]={"ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ"
,"ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾄﾞ#","
ﾚ#","ﾌｧ","ﾌｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾌｧ","ﾌｧ#","ｿ#","ﾗ
#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾌｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","."}
   70 str cc$(46)[3]={"ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ"
,"ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾚb","ﾐ
b","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ﾌｧ","ｿb","ﾗb","ｼb","
ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","."}
   80 str e$(8)[4],b$="",a$=""
  100 ca$(42)="5g+":ca$(43)="5a+":ca$(44)="5b+":ca$(45)="6c+":ca
$(46)="5r"
  110 m_init()
  120 for i=1 to 8:m_alloc(i,2000):next
  125 for i=1 to 8:m_assign(i,i):next
  128 repeat
  129 repeat
  130    input "channel no.";l
  131 until (l>=0 and l<=8)
  132 if l<>0 then {
  133 repeat
  134 io=0
  135 repeat
  136    io=i
  140    repeat
  150       a$=inkey$(0)
  160       i=i+1
  175    until a$<>""
  178 until a$<>b$ or (i-io>14 and i-io<90) or (i-io>110)
  185 if a$=chr$(13) then jmax=j else {
  200    c$(j,l)=a$:d(j,l)=i:b$=a$:print ".";
  207    m_init()
  208    for k=1 to 8
  210       f=instr(1,ba$,c$(j,k))-1
  220       if f<>-1 then {
  230          e$(k)="o"+ca$(f)
  250          m_trk(k,e$(k))
  251       }
  255    next
  260    m_play()
  270    j=j+1
  271 }
  285 until j=jmax
  286 }
  290 j=1:i=0
  295 until l=0
  300 print:print "Now repeat!"
  302 for j=1 to jmax-1
  305    a$=c$(j,l)
  306    m_init()
  308    for k=1 to 8
  310       f=instr(1,ba$,c$(j,k))-1
  320       if f=-1 then goto 380
  330       e$(k)="o"+ca$(f)
  350       m_trk(k,e$(k))
  355    next
  360    m_play()
  370    for i=1 to d(j+1,1)-d(j,1)
  380       for k=1 to 10
  390       next
  400    next
  410 next
  420 j=1:i=0:goto 128
```

多声部への発展

さあ，一晩じっくり寝て次なる目標が決まったぞ。複数の音を重ねて鳴らせるようにしてくというのである。まぁ，X68000のFM音源は放っておいても8音までは自由に使わせてくれるのだからあまりに当然の展開である。

ところがそこに大きな落とし穴があった。その第1番目は，和音を出そうとしても，

「同時に2つ以上のキーを読むということができない」

ということであった。ああ，ここにきてついに巷のキーボードに負けーてくやしい花いちもんめである。

そこでしかたがないから，各声部ごとにメロディを記憶して，あとで一度に鳴らすということにしよう。これはこれで結構使えそうである。楽譜を見て1パートごとに入力していけば，自然と全体像が見えてきてしまうというわけ。よーし，これに決めた。

第1番目の問題点と書いたが，それでは第2番目はなにか？ それは作っている最中に気づいてしまった究極の難問だった。

とにかくリスト3を見てもらおう。いままでとだいたい同じなのだが，8声まで記憶できるように配列c$やdが2次元の配列に拡張されている。130行では入力する準備としてFM音源のチャンネルNoを入力させ,lという整数変数に記憶している。これを用いて132～295行でリスト2とまったく同様に記憶を行うのである。2次元配列は，このように同じようなデータの列を複数記憶するのに便利である。参照の仕方さえ間違えなければ非常に使えるデータのかたちである。

さて，130行でチャンネルNoの代わりに0を入力すると300行へ飛ぶ。ここから420行までが，8つまでの音を同時に鳴らすルーチンなのだが，ここにきてハタと困ってしまったのである。それは，

「全音同じタイミングで鳴らせるが，それぞれ別々のリズムで鳴らすことができない」

のである。厳密にいうとできないことはないが，処理が複雑になるのでスピードが

遅くなってしまったり，逆に同時に鳴らしたいところで微妙にズレていってしまうのである。

しかし，それがわかったところでやめてしまってはいけない。とにかくやっているうちにアイデアが湧くだろうというのが暇プロの精神である。まずは，作ってしまってそれから考えるのである。それがリスト3なのである。

入力するときも他パートの音が聞きたいので208〜255行が挿入してある。この方式だと，入力するパートのリズムで他声部も鳴ってしまう。そして130行で0を入力したときはチャンネル1のリズムで全パートが鳴ってしまう。だから全パートが同じリズムの曲なら問題ないが，ズレがある曲では各声部にダミーの音を入れてやらなければならない。ちなみに休符はスペースキーなのでそれを使ってもいいし，もっとも細かく動くパートに合わせて入力してもよい。

本当はこのプログラムも真面目に改造していけばかなりいい線までいくはずで，ズレるリズムを鳴らしたり，同時の音をむりやり合わせたりすることも，各パートの音を見て変更してやればできるはずである。しかし，それはあまりにも泥臭い作業なので，もっといい方法を探すためにまた一晩じっくり眠るのである。そして，学校へ行く人は学校で，会社に行く人は会社で，さらに案を練ってから次の作業に入る。この余裕こそが明日のエネルギー源となるのである。とにかくBASICと長く付き合っていくには，こういった精神でいきたい。

深入りしすぎた

暇プログラミングをやっていると，生まれつきの性分でいつのまにか熱くなり暇が暇でなくなることがしばしばある。今回は仕事を忘れ少々深入りしすぎたというのがリスト4である。

なにせ，goto万歳などといっておきながら関数func〜endfuncを2つも作ってしまい，しかも何も値を渡さないサブルーチンとして使っているのだから大笑いである。そのうえ，コメントまでついた日には何をかいわんやである。個人のプログラムにはできるだけコメントをなくしたい(?)ものなのに，わざわざ枠で囲ってコメントを書いているとは……。やはり速いプリンタを持っていない弱みであろうか。

関数のひとつはメニュー表示のmenuである。これは“input command”で入力する文字により何が選ばれるかを表示する。

1〜8：番号で指定したチャンネルにメロディを入力。
C　：入力されたものをMMLに変換。
P　：MMLを演奏する。なかでチャンネルの選択が行える。0なら全チャンネルを演奏。
M　：入力モードを変える。最初は上書きモードだが，一度Mを入力すると最後にメロディを付け加える。
S　：データのセーブ。MMLでのセーブと生データのセーブを選ぶ。生データならあとでメロディを付け足せる。
L　：データのロード。
K　：キーボード配置の表示。

＊

ああ，手当たりしだいにコマンドを付け加えたので，コマンドごとにIF文を作ってしまうという，なんとも単純なプログラムになってしまった。しかも結構長い。その代わり，この先もう少し改造していけば楽しみなツールができた。3月号に掲載された「OPMファイル→MUSIC PRO-68Kのスコアファイル」などを利用したら楽譜の作成もあの面倒な入力作業から開放されるというものである。

千里の道も一歩から。慌てず騒がずじっくりと。これが暇プロである。これがアマグラマである。

●リスト4

```
  10 /**********************************
  20 /* program for creating MML       *
  30 /*     1990.8.22 by k.shimizu     *
  40 /**********************************
  50 /**********************************
  60 /* data initializing              *
  70 /**********************************
  80 dim str c$(100,8)[4],m$(100,8)[50],mwork$[50],mwk$[50]
  90 dim int d(100,8),jmax(8),jj(8),ipart(8),e(100,8)
 100 int j=1,f=0,i=0,k=0,l=1,iold=0,iimax=0,ll=100,ih=4,r,r0,
       r1,irflag=0,lm=16,dum=0,imode,npart=8
 110 str ba$[47]=
       "zxcvbnm,./qwertyuiop@[asdfghjkl;:1234567890-^¥ "
 120 str ca$(48)[4]={"","3c","3d","3e","3f","3g","3a","3b","4c"
,"4d","4e","4f","4g","4a","4b","5c","5d","5e","5f","5g","5a","5b
","6c","2b","3c+","3d+","3e+","3f+","3g+","3a+","3b+","4c+","4d+
","4e+","4e+","4f+","4g+","4a+","4b+","5c+","5d+","5e+"}
 130 str cb$(47)[3]={"","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ",
"ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾄﾞ#
","ﾚ#","ﾌｧ","ﾌｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾌｧ","ﾌｧ#","ｿ#"
,"ﾗ#","ﾄﾞ","ﾄﾞ#","ﾚ#","ﾌｧ","ﾌｧ#","ｿ#","ﾗ#","ﾄﾞ","ﾄﾞ#","."}
 140 str cc$(47)[3]={"","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ",
"ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ﾚ","ﾐ","ﾌｧ","ｿ","ﾗ","ｼ","ﾄﾞ","ｼ","ﾚb"
,"ﾐb","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ﾌｧ","ｿb","ﾗb","ｼb
","ﾄﾞ","ﾚb","ﾐb","ﾌｧ","ｿb","ﾗb","ｼb","ﾄﾞ","ﾚb","."}
 150 ca$(42)="5f+":ca$(43)="5g+":ca$(44)="5a+":
     ca$(45)="5b+":ca$(46)="6c+":ca$(47)="5r"
 160 str e$(8)[4],b$="",a$="",cl$="",cm$="",fil$,part$,dum$,mod
e$(2)[10]={"over write","merge"},crcl
 170 crcl=chr$(13)+chr$(10)
 180 for i=1 to 47:ca$(i)="o"+ca$(i):next
 190 /**********************************
 200 /* initialize FM                  *
 210 /**********************************
 220 m_init()
 230 for i=1 to 8:m_alloc(i,2000):next
 240 for i=1 to 8:m_assign(i,i):next
 250 for i=1 to 8:jmax(i)=0:next
 260 /**********************************
 270 /* main loop (ended by "e")       *
 280 /**********************************
 290 repeat
 300    menu()
 310    print:print "*** now in the [";mode$(imode);"] mode"
 320    print "*** please start at the 5th beep":print
 330    input "input command";cl$
 340    cl$=strupr(cl$)
 350    l=asc(cl$)-48
 360 /**********************************
 370 /* input sound by keyboard        *
 380 /**********************************
 390    if l>=1 and l<=8 then {
 400       if imode=1 then j=jmax(l)+1:i=d(jmax(l),l)-ll
 410       repeat
 420       jmax(l)=100
 430       repeat
 440          a$=inkey$(0)
 450          i=i+1
 460          if i mod ll/4=0 then beep
 470          if i=ll+ll/4+ll/lm/2-1 and j=1 then a$=" "
 480          if a$="" then for k=1 to 200:next
 490       until a$<>""
 500       if a$=chr$(13) then jmax(l)=j:d(j,l)=i else {
 510          d(j,l)=i:print ".";
 520          m_init()
 530          f=instr(1,ba$,a$)
 540          m_trk(l,ca$(f))
 550          c$(j,l)=ca$(f)
 560          m_play()
 570          j=j+1
 580       }
 590       until j=jmax(l)
 600       jmax(l)=jmax(l)-1:j=1:i=0
 610       print
 620    }
 630 /**********************************
 640 /* create MML from array d,c$     *
 650 /**********************************
 660    if cl$="C" then {
 670       print "now creating MML"
 680       for l=1 to 8
 690       for j=1 to jmax(l)+1
 700          e(j,l)=int((d(j,l)+ll/lm/2)*lm/ll)
 710       next
```

```
 720       next
 730       for l=1 to 8
 740       for j=1 to jmax(l)
 750          r=e(j+1,l)-e(j,l)
 760          mwork$=""
 770          r1=r/(4*4)
 780          r=r -r1*4*4
 790          if r=0 and r1>=1 then r=16:r1=r1-1
 800          irflag=0:mwk$=c$(j,l)
 810  if instr(1,c$(j,l),"r")<>0 then c$(j,l)="":irflag=1
 820  if r= 1 then m$(j,l)="l16"+c$(j,l)
 830  if r= 2 then m$(j,l)="l8" +c$(j,l)
 840  if r= 3 then m$(j,l)="l8"+c$(j,l)+"."
 850  if r= 4 then m$(j,l)="l4"+c$(j,l)
 860  if r= 5 then m$(j,l)="l4"+c$(j,l)+"&l16"+c$(j,l)
 870  if r= 6 then m$(j,l)="l4"+c$(j,l)+"."
 880  if r= 7 then m$(j,l)="l4"+c$(j,l)+".."
 890  if r= 8 then m$(j,l)="l2"+c$(j,l)
 900  if r= 9 then m$(j,l)="l2"+c$(j,l)+"&l16"+c$(j,l)
 910  if r=10 then m$(j,l)="l2"+c$(j,l)+"&l8"+c$(j,l)
 920  if r=11 then m$(j,l)="l2"+c$(j,l)+"&l8"+c$(j,l)+"."
 930  if r=12 then m$(j,l)="l2"+c$(j,l)+"."
 940  if r=13 then m$(j,l)="l2"+c$(j,l)+".l16"+c$(j,l)
 950  if r=14 then m$(j,l)="l2"+c$(j,l)+".&l8"+c$(j,l)
 960  if r=15 then m$(j,l)="l2"+c$(j,l)+".&l8"+c$(j,l)+"."
 970  if r=16 then m$(j,l)="l1"+c$(j,l)
 980       mwork$="l1"+c$(j,l)
 990       if r<>0 then mwork$=mwork$+"&"
1000       c$(j,l)=mwk$
1010       for k=1 to r1:m$(j,l)=mwork$+m$(j,l):next
1020       mwork$=""
1030       for i=1 to 20
1040           if irflag=1 then {
1050           if mid$(m$(j,l),i,1)="l" then mwork$=mwork$+"r"
1060           if mid$(m$(j,l),i,1)<>"l" then
                  mwork$=mwork$+mid$(m$(j,l),i,1)
1070           }
1080       next
1090       if irflag=1 then m$(j,l)=mwork$
1100       next
1110       next
1120    }
1130 /*********************************
1140 /* play MML                       *
1150 /*********************************
1160    if cl$="P" then {
1170       input "input part number(0 for all part)";part$
1180       npart=0:for i=1 to 8:ipart(i)=0:next
1190       for i=1 to 8
1200          if (instr(1,part$,itoa(i))<>0 or
                 instr(1,part$,"0")     <>0 ) then {
1210             npart=npart+1
1220             ipart(npart)=i
1230          }
1240       next
1250       print "now playing !"
1260       m_init()
1270       for l=1 to 8
1280        for j=1 to jmax(ipart(l))
1290          m_trk(l,m$(j,ipart(l)))
1300        next
1310       next
1320       m_play()
1330       j=1:i=0:l=0
1340    }
1350 /*********************************
1360 /* mode change                    *
1370 /*********************************
1380    if cl$="M" then {
1390       print "mode was changed !"
1400       imode=1-imode
1410    }
1420 /*********************************
1430 /* save data                      *
1440 /*********************************
1450    if cl$="S" then {
1460       input "choose m:MML or e:EDIT DATA";cm$
1470       input "input file name";fil$
1480       input "input part number(0 for all part)";part$
1490       npart=0:for i=1 to 8:ipart(i)=0:next
1500       for i=1 to 8
1510          if (instr(1,part$,itoa(i))<>0 or
                 instr(1,part$,"0")     <>0 ) then {
1520             npart=npart+1
1530             ipart(npart)=i
1540          }
1550       next
1560       cm$=left$(strupr(cm$),1)
1570       ifil=fopen(fil$,"c")
1580       fwrites(cm$+crcl,ifil)
1590       fwrites(itoa(npart)+crcl,ifil)
1600       for i=1 to npart
1610          fwrites(itoa(ipart(i))+crcl,ifil)
1620          fwrites(itoa(jmax(ipart(i)))+crcl,ifil)
1630          for j=1 to jmax(ipart(i))+1
1640             if cm$="M" then
                    fwrites(m$(j,ipart(i))+crcl,ifil)
1650             if cm$="E" then
                    fwrites(c$(j,ipart(i))+crcl,ifil)
1660             if cm$="E" then
                    fwrites(itoa(d(j,ipart(i)))+crcl,ifil)
1670          next
1680       next
1690       fclose(ifil)
1700       if cm$="M" then print "MML is saved in ";fil$
1710       if cm$="E" then print "EDIT DATA is saved in ";fil$
1720    }
1730 /*********************************
1740 /* load data                      *
1750 /*********************************
1760    if cl$="L" then {
1770       input "input file name";fil$
1780       ifil=fopen(fil$,"r")
1790       freads(cm$,ifil)
1800       cm$=strupr(cm$)
1810       freads(dum$,ifil)
1820       npart=atoi(dum$)
1830       for i=1 to npart
1840          freads(dum$,ifil)
1850          ipart(i)=atoi(dum$)
1860          freads(dum$,ifil)
1870          jmax(ipart(i))=atoi(dum$)
1880          for j=1 to jmax(ipart(i))+1
1890             if cm$="M" then freads(dum$,ifil)
1900             if cm$="M" then m$(j,ipart(i))=dum$
1910             if cm$="E" then freads(dum$,ifil)
1920             if cm$="E" then c$(j,ipart(i))=dum$
1930             if cm$="E" then freads(dum$,ifil)
1940             if cm$="E" then d(j,ipart(i))=atoi(dum$)
1950          next
1960       next
1970       fclose(ifil)
1980       if cm$="M" then print "MML is loaded from ";fil$
1990       if cm$="E" then print "EDIT DATA is loaded from "
                                 ;fil$
2000    }
2010 /*********************************
2020 /* display keyboard               *
2030 /*********************************
2040    if cl$="K" then {
2050       keyboard()
2060    }
2070 until cl$="E"
2080 end
2090 /*********************************
2100 /* function menu is only for      *
2110 /*      drawing main menu         *
2120 /*********************************
2130 func menu()
2140    print "*********************"
2150    print "* 1-8:sound input   *"
2160    print "*     c:create MML   *"
2170    print "*     p:play MML     *"
2180    print "*     m:mode change  *"
2190    print "*     s:save data    *"
2200    print "*     l:load data    *"
2210    print "*     k:show keyboard *"
2220    print "*     e:end the system *"
2230    print "*********************"
2240    return()
2250 endfunc
2260 /*********************************
2270 /* function keyboard is for       *
2280 /*      drawing key position      *
2290 /*********************************
2300 func keyboard()
2310    print "*****************************************"
2320    print "* ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# ﾚ# ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# *"
2330    print "* 1  2  3  4  5  6  7  8  9  0  -  ^  ¥  *"
2340    print "*                                       *"
2350    print "*  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ ﾚ  ﾐ  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ    *"
2360    print "*  q  w  e  r  t  y  u  i  o  p  @  [    *"
2370    print "*                                       *"
2380    print "*   ｼ ﾄﾞ# ﾚ# ﾌｧﾌｧ# ｿ# ﾗ# ﾄﾞﾄﾞ# ﾚ# ﾌｧ       *"
2390    print "*   a  s  d  f  g  h  j  k  l  ;  :     *"
2400    print "*                                       *"
2410    print "*    ﾄﾞ ﾚ  ﾐ  ﾌｧ ｿ  ﾗ  ｼ  ﾄﾞ ﾚ  ﾐ         *"
2420    print "*    z  x  c  v  b  n  m  ,  .  /       *"
2430    print "*****************************************"
2440    return()
2450 endfunc
2460 /*********************************
2470 /* the end of the program         *
2480 /* (lines below are for debug)    *
2490 /*********************************
2500 for i=1 to jmax(1)
2510    print d(i+1,1)-d(i,1);c$(i,1);" ";m$(i,1)
2520 next
2530 end
2540 str dum$,fil$
2550 input "input file name ";fil$
2560 ifil=fopen(fil$,"r")
2570 repeat
2580 nfil=freads(dum$,ifil)
2590 print nfil,"'";dum$;"'"
2600 until 1=0
```

［注意点］

・BEEP音が拍子の目安になります。
・BEEP音１回が４分音符にあたります。
・BEEP音１回につき４音まで入力できます。
・リズムの解像度は16分音符。
・５回目のBEEP音が曲の始めです。４回見送ってください。ここで入力がないと休符が入ります（メロディを付け足すときも同じ）。
・音を鳴らしたらすぐにキーを離してください。押し続けているとおかしくなります。音符は次の音が鳴るまで続きます（音色によって減衰する）。休符はスペースキーを押してください。
・曲の終わりはリターンキーを押してください。

ASK68K用辞書管理ユーティリティ《後編》

辞書整備応用編

Murata Toshiyuki
村田 敏幸

辞書用ユーティリティ第2弾。今回はVJE，松茸，E1などから単語を持ってきてASK68Kの辞書を強化する，というものです。そのほか，大規模テキストデータ用のツールを発表します。なお，前回のプログラムが必要ですので注意してください。

X68000とMS-DOSマシンの両方を持っている読者がどれだけいるかは疑問だが，もしも，あなたがその幸運な人（今日だけはそういうことにしておこう）のひとりで，かつ，日本語FEPとしてVJE-βか松茸V2を使っているのであれば，今回のプログラムはかなりおいしい。MS-DOSマシンなんかとは縁がなくとも，最初のX68000用市販ワープロEWを買うには買ったがいつのまにかWP.Xに舞い戻ってしまったという過去を背負った人にとってもちょっとだけおいしい。

ASK68Kの辞書メンテナンスツールお裾分け大会の第2弾は，上記3種のワープロ/日本語FEPの辞書をASK68Kに吸い上げる“CONVDIC.X”，単語ファイルをASK68Kの内部コード順にソートする“MSORT.X”，複数の辞書を切り替える常駐プログラム“CHDIC.R”の3本立てだ。辞書のコンバートについてはもう少しサポート範囲を広げるつもりはあったのだが，つもりで終わってしまった。DFJやFIXERやKatanaやEG-BridgeやWXR (WXP)のユーザーには申しわけないと思っている。

VJE，松茸，E1のユーザーも手放しでは喜べないかもしれない。辞書コンバートには多少の手間と時間が必要だ。辞書ファイルを直接ASK68K用に変換するのではなく，まず，各日本語FEPの辞書ファイルの内容を専用の辞書管理ツール（各日本語FEPについてくるもの）を利用してテキストに落としておき，それをGENDIC.Xで読み込める形式に変換して（これがCONVDIC.Xの仕事），GENDIC.XでASK68Kの辞書ファイルとして構成しなおす，という段階を踏むようになっている。すでにあるプログラムは最大限に利用しようという正しい手抜きの精神だったりする。

今回のプログラム

辞書コンバートの具体的な手順はあとで示すとして，先に各プログラムの紹介をすませておく。

●CONVDIC.X

使用法：CONVDIC［スイッチ］［入力ファイル］［［/O］出力ファイル］

各日本語FEP専用の辞書管理ツールが出力した単語ファイルをDUMPDIC.Xの出力形式（＝GENDIC.XやDELWORD.Xの入力ファイルの形式）に変換する。CONVDIC.X実行時には，元となった辞書の種類に応じて，VJEからのコンバートの場合は/B，松茸からの場合は/M，E1からの場合は/Eのスイッチをつける。

CONVDIC /B VJEB.OUT VJEB.ASK
CONVDIC /M MATU.OUT MATU.ASK
CONVDIC /E E1.OUT E1.ASK

同じノリでDICM.Xの単語一覧形式からDUMPDIC.X形式への変換もできるようにしてある。このときは/Aスイッチを使う。

CONVDIC /A ASK.OUT ASK.ASK

あといくつかのオプションがあるが，それらについてはソースか，

CONVDIC /?

によって表示されるヘルプメッセージを参照してもらいたい。

●MSORT.X

使用法：MSORT［スイッチ］［入力ファイル［……］］［/O出力ファイル］

単語ファイルをASK68Kの内部コード順でソートするフィルタ。というよりは，そういう付加機能がついた汎用のテキストソートフィルタといったほうがよいかもしれない。Human68kのSORT.Xの機能はほぼ包含している。辞書とは関係なく，SORT.Xの代わりにも使ってもらえるだろう。

MSORT.Xは単に，

MSORT

で起動すると標準入力からの入力をソートして標準出力に書き出す。

MSORT FILE

のようにファイル名を指定すれば，入力がそちらに切り替わる。ファイル名は，複数並べてもよいし，ワイルドカードを使って指定してもよい。

MSORT FILE1 FILE2 FILE3
MSORT *.DAT

ソート結果をファイルに残したいときには，出力をリダイレクトするか，/Oスイッチで出力ファイル名を指定する。

MSORT FILE >OUTPUT
MSORT FILE /OOUTPUT

MSORT.Xは入力ファイルが巨大でメモリに収まりきらないときには途中経過を一時的な作業用ファイルに書き出し，あとでマージする。ファイルがメモリに収まらないと止まってしまうSORT.Xに比べればずっと頑丈だ。

特に指定しないとMSORT.Xは5000行単位でソートを実行する。メモリにゆとりがあるときには/Lスイッチで処理する行数を大きくしておくと若干実行速度の向上が期待できる（一時ファイルを作らずにすませられれば最高速)。一時ファイルを作成するドライブ（環境変数tempで指定する）にはフロッピーディスクよりもハードディスク，ハードディスクよりもRAMディスクを割り当てたほうが望ましいのはいうまでもない。が，一時ファイルを作成するドライブの空き容量でソートできるファイルの最大サイズが決まるということを付け加えておく。

実行速度といえば，一時ファイルの入出力を行うというハンデがあるにもかかわらず，多くの場合MSORT.XはSORT.Xよりも高速に動作する。ソートするファイルが大きくなるほどその差は広がり，MSORT.Xは2Mバイト強のファイルを30分かからずにソートする（GCCでコンパイルし，一時ファイルはハードディスク上に作成した）が，SORT.Xは1Mバイト弱のファイルのソートに一晩かかる。MSORT.Xはなんの工夫もなく

クイックソートしているだけなのだが，SORT.Xはそれに輪をかけて工夫が足りないらしい。

MSORT.Xにはいくつかのスイッチがある。辞書に関係あるのが/Dスイッチだ。/Dを指定すると入力ファイルを単語ファイルとみなし，“読み”のみを比較対象としてASK68Kの内部コード順にソートする。先月も触れたようにGENDIC.Xは入力となる単語ファイルがすでにASK68Kの辞書内部での順番になっているものと仮定して処理を行うから，DUMPDIC.XやDICM.Xの出力以外のものをGENDIC.Xに与えるときには，あらかじめMSORT.Xで/Dスイッチをつけてソートしておくことが望ましい。

残るスイッチについても簡単に触れておく。/Uをつけると同一の行が複数続くときに1行だけを残すようになる。/Mはソートをせずに複数ファイルのマージだけを行うことを指定する。入力ファイルがソート済みであれば，マージされて出てきたファイルも正しくソートされている。入力ファイルがソートされていないときは，たぶん意味のない結果が得られる。あと，/I，/K，/RはSORT.Xと同じ意味を持つ。Human68kのマニュアルを参照のこと。

●CHDIC.R

使用法：CHDIC 辞書ファイル［……］

複数の辞書ファイルを切り替えて使うための常駐プログラムだ。辞書のコンバートとは直接の関係はない。一度実行するとメモリ上に居座り，以後，XF5を押すたびにメイン辞書を切り替える。もちろん，ASK68K上ではF8キーを押してから辞書ファイル名を入力すれば辞書を変更できるわけだが，CHDIC.Rはその手間を省くことだけを目的としたプログラムなのだ。しかも，インチキをしているので，WP.X上では使用できない。ま，半分ぐらいのところまでは冗談プログラムだと思ってもらったほうがよいかもしれない。

辞書ファイルの切り替えにはASK68Kの内部ファンクションを使わずに，いわゆるKEY0っぽい処理で逃げている。つまり，XF5が押されるとF8＋辞書ファイル名が入力されたことにしてシステムをだますわけだ。

当初はASK68Kの内部ファンクションを使って辞書を切り替えていたのだが，どうも効果が現れない。たぶん，メモリ上にすでに読み込んであった前の辞書の一部が残っていて，その部分についてはディスクを読みにいかないのだと思う。ASK68K内部のワークエリアを書き換えるような汚いことをすればなんとかなるのだろうが，そっちの方向は避けたかった。で，苦肉の策がKEY0もどきだ。

使用感はそれほど悪くはないが，よくもない。ASK68Kが辞書ファイル名変更時のキー入力部分にウエイトを入れているらしく，XF5を押してから辞書が切り替わるまでに2〜3秒かかる。また，あくまでキー入力をシミュレーションしているだけなので，かな漢字変換途中の文字が残っていたりすると，辞書ファイル名がそのまま変換行に流れ込むといったことも起こる。なかなかわがままなプログラムになってしまった。

慣例に従い，/Rスイッチをつけるとすでに常駐しているCHDIC.Rがあれば，その常駐を解除する。ただし，安全のため，CHDIC.Rがフックしたベクタがさらに別のプログラムによってフックされているときには常駐解除できないようになっている。常駐プログラムは常駐させたのと逆の順序で解放するのが鉄則だ。

なお，ASK68K Ver.2.0を使っていて，辞書ファイルの変更をF8以外のキーに割り当てている人にはプログラムの一部を書き換えることで対処してもらう。詳しくはプログラムの入力方法の項で触れる。

辞書コンバートの実際

では，辞書コンバートの手順を見ていこう。

●ステップ1

まず，とにかく各日本語FEPの辞書をテキストファイルに落とす。VJEの場合はVX.EXE，松茸の場合は新松の辞書管理メニュー，E1の場合はEDICM.Xを使う。細かな手順は次のとおり。

・VJE-βの場合

1) VX.EXEを以下のようにして起動する。

VX－s－g disp 辞書ファイル名 > 出力ファイル名

ASK68Kの辞書構造

どうもライター間の意志の疎通がうまくいっていなかったようで，先月の中野氏の記事中で以前本誌に掲載されたままのASK68Kの辞書解析結果が掲載されていた。そこで，ここではあの図表に対する補足ということで話を進めさせてもらう。以下の文中では図の番号は先月号54ページのもので，用語もあそこにあるものを使う。とりあえず9月号を手元に用意してほしい。

まず，ASK68Kの辞書ファイルの大雑把な構造だ。図1ではオフセット$1E00_H$〜$1FFF_H$がフラグ領域かもしれないというコメントがあるが，フラグ領域はたぶん$1F00_H$〜$1FFF_H$だと思われる。その計算でいくと，インデックス領域は992ブロック分となり，これがASK68Kの辞書ファイルの最大の大きさとなる。

フラグ領域は1バイトが見出し語コード（図3)のひとつに対応し，そのコードから始まる単語が辞書ファイル中に存在すれば01_H，なければ00_Hになる。たとえば，“ダ”で始まる単語が存在すればオフセット$1FC0_H$が01_Hになる。ところで，00_H〜$1F_H$で始まる単語は辞書中には存在しないはずだから，フラグ領域は$1F20_H$以降と考えるべきなのかもしれない(そうすると辞書ファイルの最大サイズはもう4ブロック分大きくなる)。が，DUMPDIC.XやGENDIC.Xではとりあえず安全のため$1F00_H$を境界とみなしている（256バイトという区切りのよさに屈服してしまったのだ)。

ちなみに，DICM.Xではこの境界のチェックをまったくしていないようで，辞書を再編成したりマージしたりして辞書が大きくなると平気な顔でインデックス領域がフラグ領域以下を浸食して辞書を破壊する（おいおい)。

フラグ領域中，特にオフセット$1FFF_H$はASK68Kの辞書ファイルのバージョンを示す。ASK68K Ver.1.0用の辞書では00_H，Ver.2.0用の辞書ではFF_Hとなる。Ver.2.0用の辞書では辞書ファイルの末尾に8Kバイトの意味深なデータが追加されている。たぶん，変換効率を上げるための補助データなのだろう。辞書に単語を登録したり削除したりしても内容に変化は見られなかった。で，妙なことにDICM.Xで辞書を再編成したり，マージしたりするとこの部分は消えてなくなってしまう(こらこら)。

続いて辞書の1ブロックの構造（図2）だ。ほとんど図2のとおりだが，DICM.Xの動作を見ていると，各ブロックの末尾2バイトは0000_Hでなければならないことになっているふしがある。また，ごく一部の特殊な単語に限り品詞コードが2バイトになる場合があるようだ。ASK68Kはデバイスドライバ本体の中に辞書ファイルとほとんど同じ形式のデータ(主としてひらがなだけの単語）数百語分を抱え込んでいて（こらこら)，これが辞書学習の結果サブ辞書ににじみ出てくるらしい。

最後に，図3の見出し語コードでは90_H台に以下の8つが割り当てられていることがわかっている。

98_H　。(句点)
99_H　「
$9A_H$　」
$9B_H$　、(読点)
$9C_H$　・(中点)
$9D_H$　ー（長音記号)
$9E_H$　゛（濁点)
$9F_H$　゜（半濁点)

2) 画面には何も表示されないが，構わず出力開始位置と終了位置の“読み”を入力する（それぞれ最後にリターンキーを押す）。
3) もう一度リターンキーを押すと処理が開始されディスクが回り出す。
4) ディスクが止まるのを待って，リターンキーを押すとVX.EXEから抜ける。
5) VX.EXEの出力には前後にゴミが残っているのでエディタなどで削る。
・松茸V2の場合
1) 新松を起動し，メニューから“辞書管理”を選ぶ。
2) “品詞制限”で全品詞を処理対象にしておく。
3) “辞書一覧”を選ぶ。
4) サブメニューの“印刷”で以下のように設定する。

印刷	しない
頁替え	しない
テキストファイル出力	する
区切り文字	半空

表1

	convdic.x	msort.x
convdic.c	○	
elconv.c	△	
matuconv.c	△	
vjeconv.c	△	
msort.c		○
wild.c		○
(parseline.c)	○	
(strfunc.c)	○	○
(misc.c)	○	○
(vfprintf.s)	○	○
(mydef.h)	○	○
(misc.h)	○	○
(strfunc.h)	○	○
(myerror.h)	○	○
(dictools.h)	○	
wild.h		○

図1 CHDIC.Rの環境別修正点

・辞書ファイルの機能を……
1) ファンクションキー（単独）に割り当てた場合
000E 6A → 62+n（nはFnのn）
2) SHIFT+ファンクションキーに割り当てた場合
000C 00 → 70
000E 6A → 62+n（nはFnのn）
0010 00 → F0
3) CTRL+ファンクションキーに割り当てた場合
000C 00 → 71
000E 6A → 62+n（nはFnのn）
0010 00 → F1
・いつも使っている変換モードが……
1) 辞書先読みなしの一括変換の場合
0017 02 → 01
2) 辞書先読みありの一括変換の場合
変更の必要なし
3) 逐次変換の場合
0017 02 → 03

5) 出力ファイル名，出力範囲，出力品詞を指定する
・E1の場合
1) EDICM.Xを起動する。
2) メニューから“拡張機能”，さらに“単語一覧”を選択する。
3) 出力ファイル名，出力範囲，出力品詞を指定する。

いずれの場合でも，辞書の全内容をテキストに落とすと2Mバイトを越える巨大なファイルが作成される。フロッピーディスク上で作業を行うときには読みの範囲や品詞別に2回以上に分割して出力する必要がある。この場合ステップ2以下の操作は各ファイルごとに行い，ASK68Kの辞書にした時点でDICM.Xでマージすること。

VJEや松茸の場合は，ファイルをX68000に持ってこなければ話にならないわけだが，ご存じのようにHuman68kではMS-DOSの標準的な5インチ2HDディスクの読み書きができるから，基本的にはガッチャンとイジェクトしたディスクをジーコーとX68000に挿入すればファイルが転送できる。不幸にして，あなたのMS-DOSマシンのディスクドライブが3.5インチだったり，5インチでも旧型の2DD専用だったりした場合は，RS-232CクロスケーブルでX68000とつないで転送することになるだろう。

●ステップ2

上で作成した単語ファイルは各日本語FEP付属ツールの独自形式であり，まだGENDIC.XでASK68Kの辞書ファイルに変換できるかたちにはなっていない。各行のフォーマットは異なるし，品詞名も日本語FEPごとに微妙に違う。そこで，前述したように変換元の辞書に応じて/V，/M，/EいずれかのスイッチをつけてCONVDIC.XによりGENDIC.Xが読み込める形式に変換する。

●ステップ3

CONVDIC.Xを通してもなおクリアできていない問題がある。単語ファイル内の単語の順番だ。MSORT.Xで/Dスイッチをつけて単語ファイルをソートするのがベストの解決法となる。ただ，単語の順番が多少狂ったままでもあとでなんとかできないこともないから，ソートを省略するという道も残されてはいる（あまり勧められないが）。

●ステップ4

ここまでくれば，あとは単語ファイルをGENDIC.Xにかけるだけだ。警告メッセージがだいぶ出るはずだが，気にすることはない。警告は“二重登録”がほとんどで，ときおり“無効な品詞名”が混じる程度だと思う。どちらもCONVDIC.Xが品詞名のつじつま合わせを行った正しい結果だ。

CONVDIC.XはASK68Kにコンバートしても意味のない単語にはGENDIC.Xの処理からはずすためにそれなりの印をつけ，それが“無効な品詞名”の警告として現れる。また，品詞の種類によっては1語が異なる品詞の2語に展開されることがあり，その過程で元々辞書にあった単語とダブれば“二重登録”になる。

これら以外にも，特にVJEからコンバートした場合はVX.EXEが出力した単語ファイルの中に最初から重複があるため山ほど“二重登録”の警告が出る。VX.EXEの品詞区分はVJE内部での品詞分類より大雑把で，内部では微妙に異なる品詞を同一の品詞名として出力していると考えられる。

さて，ステップ3のソートをサボった場合は“単語の並び順が異なる”という警告が出ることになる。先月の60ページ3段目上から6行目以下に示した方法で対処するように（エラーがなくなるまで繰り返すんだよ）。

●ステップ5

コンバートした辞書はX68K_M.DICにDICM.Xでマージして使うもよし，単体でX68K_M.DICの代わりに使うもよし，読者のアイデア次第でどうにでもなる。ここでは2，3のコツと注意点を挙げるにとどめよう。

X68K_M.DICにマージして使うときにはASK68Kの辞書ファイルの大きさの制限に気をつけること。大体1Mバイトをちょっと越えたあたりに限界がある（怖いことにDICM.Xはマージ時に辞書ファイルサイズが上限を越えても教えてくれない）。

コンバートした辞書をごっそりマージするよりは必要な品詞だけを選んでX68K_M.DICに付け足してやるほうが賢いと思う。名詞とサ行変格複合名詞，形容動詞複合名詞あたりが狙い目だ。話が前後するが，最初からこれらの品詞だけをコンバートすれば作業の手間と時間の節約にもなる。

あと，各辞書の癖や傾向を見極めることも大切だ。たとえば，松茸の辞書では“強”という漢字が“つよい”という訓読みで単漢字として辞書登録されている。これはなかなか強力だが，うかつにメイン辞書にマージしてしまうと変換効率を落としてしまうことにもなりかねない。

参考までに僕の辞書が今現在どうなっているか話しておこう。基本的にはX68K_M.DICと松茸の辞書を合成したものから，単

漢字，地名，人名，郵便番号辞書をごっそり削ったものをメイン辞書にしている。が，さすがに単漢字がないと不便だし，そのほかの部分も必要な場合があるので，これらは別の辞書ファイルにまとめ，CHDIC.Rで切り替えられるようにしてある。単漢字を別辞書に分けたのは上で述べた松茸対策だったのだが，ASK68Kお得意の“単漢字並べ当て字変換”がなくなった分，変換効率がよくなったような気がした。あわててサブ辞書中の単漢字も全部削ったのはいうまでもない。

入力方法

CHDIC.R以外は先月同様Cで分割コンパイルを前提に記述してある。コンパイル方法については先月号を参照してもらうとして，表1にCONVDIC.XとMSORT.Xを生成するのにどのファイルが必要かだけまとめておく。カッコでくくってあるファイルは先月掲載したものだ。ここで，△がついたE1CONV.C，MATUCONV.C，VJECONV.Cは必要な人だけ入力すればよいようになっている。自分に関係のない日本語FEP用のソースは入力しなくてもよい。ただし，その際にはCONVDIC.Cの24〜26行のマクロ定義部分を注釈に示したように変更すること。

CHDIC.Rだけはダンプリストのかたちで提供しているので，リスト8を6月号の付録ディスクにも収録されたダンプ入力ツールMAC.Xで入力する。このとき，ASK68KのVer.2.0のユーザーは環境設定に応じて，図1に示す該当箇所を修正する必要がある。この図は，

0017 02 → 03

とあったら，CHDIC.Rの0017Hバイト目を02Hから03Hに変更する，と読む。

運悪くXCのVer.2.0の製品直前版が編集部に届いていたので念のためそちらでもコンパイルできるかどうか試してみた。なんとかコンパイルはできるようだが，動作確認はまだしていない。なお，先月のVFPRINTF.S関係の部分はver2.0では使用できず，必要ないので取りはずす。ただしprintfがエラーを返さないので注意すること。またリスト11で二重定義のエラーが出るはずなので12〜15行の定義部を削除してほしい。

＊

先月と今月のツールを使えばASK68Kの辞書はかなり洗練され，充実するはずだ。変換効率もだいぶ改善されるだろう。とにかく，道具は提供した。あとは，ヒット率100％を目指して（絶対無理だけど）少しでも自分にとって使いやすいようにあれこれと辞書をいじくり回してみてほしいと思う。

リスト1　convdic.c

```
  1: /*
  2:         CONVDIC.X
  3:         単語ファイルレベルでの辞書コンバータ
  4:
  5:         convdic.c parseline.c wclass.c
  6:         ( elconv.c matuconv.c vjeconv.c )
  7:         strfunc.c misc.c vfprintf.s
  8:         mydef.h strfunc.h misc.h myerror.h dictools.h
  9:         doslib.a iocslib.a
 10: */
 11:
 12: #include    "mydef.h"
 13: #include    <stdio.h>
 14: #include    <stdlib.h>
 15: #include    <ctype.h>
 16: #include    <string.h>
 17: #include    <jfctype.h>
 18: #include    <jstring.h>
 19: #include    "misc.h"
 20: #include    "strfunc.h"
 21: #include    "myerror.h"
 22: #include    "dictools.h"
 23:
 24: #define     _E1     1   /*E1からの変換をしない場合は0*/
 25: #define     _MATU   1   /*松茸V2からの変換をしない場合は0*/
 26: #define     _VJE    1   /*VJE-βからの変換をしない場合は0*/
 27:
 28: PACKEDSTR progname = "CONVDIC";
 29: PACKEDSTR usagemes =
 30: "機　能：各種日本語入力FEPの単語ファイル¥
 31: をDUMPDIC形式に変換します¥n¥
 32: 使用法：%s [スイッチ] [入力ファイル ] [[/O]出力ファイル]¥n¥
 33: ¥t/A¥tASK68K (DICM.X) の辞書一覧形式から (デフォルト) ¥n¥
 34: ¥t/E¥tE1(EDICM.X)の辞書一覧形式から¥n¥
 35: ¥t/M¥t松茸V2 (新松) の辞書一覧形式から¥n¥
 36: ¥t/B¥tVJE-β (VX.EXE) の辞書一覧形式から¥n¥
 37: ¥n¥
 38: ¥t/H¥t見出し語を半角カタカナで出力する¥n¥
 39: ¥t/G¥t見出し語を全角ひらがなで出力する¥n¥
 40: ¥t/K¥t見出し語を全角カタカナで出力する (デフォルト) ¥n¥
 41: ¥t/V¥t実行経過を報告する¥n";
 42:
 43: #if _E1
 44:     void elconv( FILE *, FILE * );
 45: #else
 46:     #define elconv      dummy
 47: #endif
 48:
 49: #if _MATU
 50:     void matuconv( FILE *, FILE * );
 51: #else
 52:     #define matuconv    dummy
 53: #endif
 54:
 55: #if _VJE
 56:     void vjeconv( FILE *, FILE * );
 57: #else
 58:     #define vjeconv     dummy
 59: #endif
 60:
 61: static boolean hankaku = FALSE;
 62: static boolean hiragana = FALSE;
 63: boolean verbose = FALSE;
 64:
 65: static void windup()
 66: {
 67:     void B_COLOR( int );
 68:     B_COLOR( 3 );
 69: }
 70:
 71: static void putword( fp, index, word, class )
 72: FILE *fp;
 73: STRPTR index, word, class;
 74: {
 75:     STR index0, word0;
 76:
 77:     setindex( index0, index, hankaku, hiragana );
 78:     settab( strcpy( word0, word ), 3 * 8 );
 79:     if ( fprintf( fp, "%s%s%s¥n", index0, word0, class ) == E
    OF )
 80:         diskfull( fp );
 81: }
 82:
 83: static boolean classconv( class, convtbl )
 84: STRPTR class, *convtbl;
 85: {
 86:     for ( ; *convtbl != NULL; convtbl++ ) {
 87:         if ( !strcmp( *convtbl++, class ) ) {
 88:             strcpy( class, *convtbl );
 89:             break;
 90:         }
 91:     }
 92:     return ( *class != '%' );
 93: }
 94:
 95: void xputword( fp, index, word, class, convtbl )
 96: FILE *fp;
 97: STRPTR index, word, class, *convtbl;
 98: {
 99:     STR index0, word0;
100:     STRPTR p;
101:     void putword( FILE *, STRPTR, STRPTR, STRPTR );
102:
103:     if ( classconv( class, convtbl ) ) {
104:         putword( fp, index, word, class );
105:     } else {
106:         for ( p = strtok( class, "/" );
107:             p != NULL; p = strtok( NULL, "/" ) ) {
108:             sprintf( index0, p, index );
109:             sprintf( word0, strtok( NULL, "/" ), word );
110:             putword( fp, index0, word0, strtok( NULL, "/" ) )
    ;
111:         }
112:     }
113: }
114:
115: static STRPTR askconvtable[] = {
116: #if 0
117:     "名詞",     "形容動詞複合名詞",
118:     "形容詞",
119:         "%s/%s/形容詞/%sサ/%sさ/名詞/%sゲ/%sげ/形容動詞複合名
    詞",
120: #endif
121:     NULL
122: };
123:
124: static void askconv( sourfp, destfp )
125: FILE *sourfp, *destfp;
126: {
127:     STR linbuf, index, word, class;
128:     int KEYSNS( void );
129:
130:     for ( *index = '¥0'; fgets( linbuf, NCHAR, sourfp ) != NU
    LL; ) {
131:         KEYSNS();
132:         if ( verbose )
```

```
133:            fprintf( stderr, "¥033[31m%s¥033[m", linbuf );
134:        if ( parseline( linbuf, index, word, class ) )
135:            xputword( destfp, index, word, class, askconvtabl
e );
136:    }
137: }
138:
139: static void dummy(){}
140:
141: void main( argc, argv )
142: int argc;
143: STRPTR *argv;
144: {
145:     FILE *sourfp, *destfp;
146:     STRPTR sourfile = NULL;
147:     STRPTR destfile = NULL;
148:     STR linbuf;
149:     STR index, word, class;
150:     boolean dflt = TRUE;
151:     void (*convfunc)() = NULL;
152:
153:     for ( ; --argc; ) {
154:         if ( strchr( SWITCH_DELIMITER, **++argv ) != NULL ) {
155:             unsigned int sc;
156:             sc = *++*argv;
157:             switch ( tolower( sc ) ) {
158:                 case '¥0':
159:                     setptr( &sourfile, stdin ); /*dummy*/
160:                     break;
161:                 case 'a':
162:                     setptr( &convfunc, askconv );
163:                     break;
164:                 case 'e':
165:                     setptr( &convfunc, elconv );
166:                     break;
167:                 case 'm':
168:                     setptr( &convfunc, matuconv );
169:                     break;
170:                 case 'b':
171:                     setptr( &convfunc, vjeconv );
172:                     break;
173:                 case 'o':
174:                     setptr( &destfile, ++*argv );
175:                     break;
176:                 case 'h':
177:                     setflag( &hankaku );
178:                     break;
179:                 case 'g':
180:                     setflag( &hiragana );
181:                     break;
182:                 case 'k':
183:                     if ( hiragana | hankaku )
184:                         usage();
185:                     break;
186:                 case 'v':
187:                     setflag( &verbose );
188:                     break;
189:                 default:
190:                     usage();
191:                     break;
192:             }
193:         } else if ( sourfile == NULL ) {
194:             sourfile = *argv;
195:             dflt = FALSE;
196:         } else if ( destfile == NULL ) {
197:             destfile = *argv;
198:         } else {
199:             usage();
200:         }
201:     }
202:
203:     if ( convfunc == NULL )
204:         convfunc = askconv;
205:
206:     resetstdin();
207:     if ( dflt )
208:         sourfp = stdin;
209:     else if ( ( sourfp = fchkopen( sourfile, "r" ) ) == NULL
)
210:         fatal_error( ROPENERRMES, sourfile );
211:     destfp = setstdout( destfile );
212:
213:     breakset( windup );
214:
215:     (*convfunc)( sourfp, destfp );
216:
217:     fclose( sourfp );
218:     fclose( destfp );
219:
220:     exit( EXIT_SUCCESS );
221: }
```

リスト2 elconv.c

```
 1: #include     "mydef.h"
 2: #include     <stdio.h>
 3: #include     <stdlib.h>
 4: #include     <ctype.h>
 5: #include     <string.h>
 6: #include     "strfunc.h"
 7: #include     "dictools.h"
 8:
 9: static STRPTR elconvtable[] = {
10:       "五段カ行",           "カ五動詞",
11:       "五段ガ行",           "ガ五動詞",
12:       "五段サ行",           "サ五動詞",
13:       "五段タ行",           "タ五動詞",
14:       "五段ナ行",           "ナ五動詞",
15:       "五段バ行",           "バ五動詞",
16:       "五段マ行",           "マ五動詞",
17:       "五段ラ行",           "ラ五動詞",
18:       "五段ワ行",           "ワ五動詞",
19:       "カ変",               "カ変動詞",
20:       "サ変",               "サ変動詞",
21:       "サ変名詞",           "サ変複合名詞",
22:       "単漢",               "単漢字",
23:       "形容動詞(ﾀﾞ型)",     "形容動詞複合名詞",
24:       "数詞",               "数字*",
25:       "一段(非体言)",       "上・下一段動詞",
26:       "姓",                 "人名(姓)",
27:       "名前",               "人名(名)",
28:       "地名(県市区名)",     "地名",
29:       "会社名、官庁名",     "団体名",
30:       "固有名詞",           "物の名称",
31:       "代名詞",             "名詞",
32:       "副詞(普通)",         "副詞",
33:       "副詞(ﾄ,ﾆ型)",        "形容動詞複合名詞",
34:       "接頭語(普通)",       "名詞",
35:       "接頭語(数詞)",       "名詞",
36:       "接尾語(普通)",       "接尾語",
37:       "接尾語(人名)",       "接尾語",
38:       "接尾語(地名)",       "接尾語",
39:       "接尾語(数詞)",       "数詞",
40:
41:       "ザ変",
42:            "%s/%s/サ変動詞/%sジ/%sじ/上・下一段動詞",
43:       "一段(体言)",
44:            "%s/%s/名詞/%s/%s/上・下一段動詞",
45:       "形容動詞(ﾀﾙ,ﾄ型)",
46:            "%s/%s/形容動詞複合名詞/%sタル/%sたる/連体詞",
47:
48:       NULL
49: };
50:
51: void elconv( sourfp, destfp )
52: FILE *sourfp, *destfp;
53: {
54:     STR linbuf, index, word, dmy, class;
55:     int KEYSNS( void );
56:     extern boolean verbose;
57:
58:     for ( *index = '¥0'; fgets( linbuf, NCHAR, sourfp ) != NU
LL; ) {
59:         KEYSNS();
60:         if ( verbose )
61:             fprintf( stderr, "¥033[31m%s¥033[m", linbuf );
62:         *word = *class = '¥0';
63:         if ( isspace( *linbuf ) ) {
64:             sscanf( linbuf, "%s%s%s", word, dmy, class );
65:         } else {
66:             sscanf( linbuf, "%s%s%s%s", index, word, dmy, cla
ss );
67:         }
68:         if ( !strcmp( word, "※" ) )
69:             strcpy( word, index );
70:         strzenkata( index, index );
71:         if ( !strcmp( word, "*" ) )
72:             strcpy( word, index );
73:
74:         if ( *index !='¥0' && *word != '¥0' && *class != '¥0'
)
75:             xputword( destfp, index, word, class, elconvtable
);
76:     }
77: }
```

リスト3 matuconv.c

```
 1: #include     "mydef.h"
 2: #include     <stdio.h>
 3: #include     <stdlib.h>
 4: #include     <ctype.h>
 5: #include     <string.h>
 6: #include     <jstring.h>
 7: #include     "strfunc.h"
 8: #include     "dictools.h"
 9:
10: static STRPTR matuconvtable[] = {
11:       "カ行5段",       "カ五動詞",
12:       "ガ行5段",       "ガ五動詞",
13:       "サ行5段",       "サ五動詞",
14:       "タ行5段",       "タ五動詞",
15:       "ナ行5段",       "ナ五動詞",
16:       "バ行5段",       "バ五動詞",
17:       "マ行5段",       "マ五動詞",
18:       "ラ行5段",       "ラ五動詞",
19:       "ワァ行5段",     "ワ五動詞",
20:       "サ変動詞",      "サ変複合名詞",
21:       "サ変動非名詞", "サ変動詞",
22:       "1段非名詞",     "上・下一段動詞",
23:       "ーーがる",      "形容詞",
24:       "形容動詞", "形容動詞複合名詞",
25:       "形容動詞2",     "形容動詞複合名詞",
26:       "形容動詞3",     "形容動詞複合名詞",
```

```
27:         "形容動詞5",     "形容動詞複合名詞",
28:         "一般名詞",      "名詞",
29:         "接尾辞",        "接尾語",
30:         "接続・感動詞",  "接続詞",
31:         "コソアド副詞",  "副詞",
32:         "ンナ連体詞",    "連体詞",
33:         "人名",          "人名(姓)",
34:         "固有名詞",      "物の名称",
35:
36:         "ザ変動詞",
37:             "%s/%s/サ変動詞/%sジ/%sじ/上・下一段動詞",
38:         "1段名詞",
39:             "%s/%s/名詞/%s/%s/上・下一段動詞",
40:         "形容動詞4",
41:             "%s/%s/形容動詞複合名詞/%sタル/%sたる/連体詞",
42:         "−−げ",
43:             "%sゲ/%sげ/形容動詞複合名詞/%sゲ/%s気/形容動詞複合名
    詞",
44:         "−−め",
45:             "%sメ/%sめ/形容動詞複合名詞/%sメ/%s目/形容動詞複合名
    詞",
46:         "かすける接続",
47:             "%sカ/%sか/サ五動詞/%sケ/%sけ/上・下一段動詞",
48:         "かるける接続",
49:             "%sカ/%sか/ラ五動詞/%sケ/%sけ/上・下一段動詞",
50:         "がるげる接続",
51:             "%sガ/%sが/ラ五動詞/%sゲ/%sげ/上・下一段動詞",
52:         "すれる接続",
53:             "%s/%s/サ五動詞/%sレ/%sれ/上・下一段動詞",
54:         "せる接続",
55:             "%sセ/%sせ/上・下一段動詞/%s/%s/上・下一段動詞",
56:         "まるめる接続",
57:             "%sマ/%sま/ラ五動詞/%sメ/%sめ/上・下一段動詞",
58:         "らすれる接続",
59:             "%sラ/%sら/サ五動詞/%sレ/%sれ/上・下一段動詞",
60:         "わるえる接続",
61:             "%sワ/%sわ/ラ五動詞/%sエ/%sえ/上・下一段動詞",
62:         "むまる接続",
63:             "%s/%s/マ五動詞/%sマ/%sま/ラ五動詞",
64:
65:     NULL
66: };
67:
68: void matuconv( sourfp, destfp )
69: FILE *sourfp, *destfp;
70: {
71:     STR linbuf, index, word, class;
72:     WCHAR c;
73:     STRPTR p;
74:     int KEYSNS( void );
75:     extern boolean verbose;
76:
77:     for ( *index = '¥0'; fgets( linbuf, NCHAR, sourfp ) != NU
LL; ) {
78:         KEYSNS();
79:         if ( verbose )
80:             fprintf( stderr, "¥033[31m%s¥033[m", linbuf );
81:         *word = *class = '¥0';
82:         sscanf( linbuf, "%s%s%s", index, word, class );
83:         if ( ( *word == '¥"' )
84:             && ( p = strrchr( word, '¥"' ) ) && ( *++p == '¥0
' ) ) {
85:             *--p = '¥0';
86:             strcpy( word, word + 1 );
87:         }
88:         strzenkata( index, index );
89:
90:         if ( ( *index & *word & *class ) != 0 ) {
91:             c = *( ( WCHAR * ) word );
92:             if ( ( c > 0x84be && c < 0x889f ) || c > 0xeaa2 )
93:                 strcat( class, "/JIS規格外文字" );
94:             if ( strlen( word ) > 2 && jstrchr( word, '$' )
)
95:                 strcat( class, "/松茸特有" );
96:
97:             xputword( destfp, index, word, class, matuconvtab
le );
98:         }
99:     }
100: }
```

リスト4 vjeconv.c

```
 1: #include    "mydef.h"
 2: #include    <stdio.h>
 3: #include    <stdlib.h>
 4: #include    <ctype.h>
 5: #include    <string.h>
 6: #include    "strfunc.h"
 7: #include    "dictools.h"
 8:
 9: static STRPTR vjeconvtable[] = {
10:         "カ五",         "カ五動詞",
11:         "カ変",         "カ変動詞",
12:         "ガ五",         "ガ五動詞",
13:         "サ五",         "サ五動詞",
14:         "サ変",         "サ変動詞",
15:         "タ五",         "タ五動詞",
16:         "ナ五",         "ナ五動詞",
17:         "バ五",         "バ五動詞",
18:         "マ五",         "マ五動詞",
19:         "ラ五",         "ラ五動詞",
20:         "ワ五",         "ワ五動詞",
21:         "一段",         "上・下一段動詞",
22:         "感動",         "感動詞",
23:         "企業",         "団体名",
24:         "形動",         "形容動詞",
25:         "形容",         "形容詞",
26:         "固名",         "物の名称",
27:         "助数",         "数詞",
28:         "人姓",         "人名(姓)",
29:         "人名",         "人名(名)",
30:         "数詞",         "数字*",
31:         "接続",         "接続詞",
32:         "接頭",         "名詞",
33:         "接尾",         "接尾語",
34:         "単漢",         "単漢字",
35:     "名サ",         "サ変複合名詞",
36:     "名形",         "形容動詞複合名詞",
37:     "名詞",         "名詞",
38:     "連体",         "連体詞",
39:
40:     NULL
41: };
42:
43: void vjeconv( sourfp, destfp )
44: FILE *sourfp, *destfp;
45: {
46:     STR linbuf, index, word, class;
47:     WCHAR c;
48:     int KEYSNS( void );
49:     extern boolean verbose;
50:
51:     for ( *index = '¥0'; fgets( linbuf, NCHAR, sourfp ) != NU
LL; ) {
52:         KEYSNS();
53:         if ( verbose )
54:             fprintf( stderr, "¥033[31m%s¥033[m", linbuf );
55:         sscanf( linbuf, "%s%s%s", index, word, class );
56:         strzenkata( index, index );
57:         class[6] = '¥0';
58:         c = *( ( WCHAR * ) word );
59:         if ( ( c > 0x84be && c < 0x889f ) || c > 0xeaa2 )
60:             strcat( class, "/JIS規格外文字" );
61:         if ( *index !='¥0' && *word != '¥0' && *class != '¥0'
 )
62:             xputword( destfp, index, word, class + 2, vjeconv
table );
63:     }
64: }
```

リスト5 msort.c

```
 1: /*
 2:         MSORT.X
 3:         テキストをソートするフィルタ
 4:
 5:         msort.c wild.c strfunc.c misc.c
 6:         mydef.h strfunc.h misc.h myerror.h
 7:         doslib.a iocslib.a
 8: */
 9:
10: #include    "mydef.h"
11: #include    <stdio.h>
12: #include    <stdlib.h>
13: #include    <ctype.h>
14: #include    <string.h>
15: #include    <doslib.h>
16: #include    <limits.h>
17: #include    "misc.h"
18: #include    "strfunc.h"
19: #include    "wild.h"
20: #include    "myerror.h"
21:
22: #define     NLINE       5000
23: #define     LINBUFSIZ   4096
24: #define     NMERGE      10
25: #define     NTEMP       0x10000000
26: #define     RSV         32768
27:
28: PACKEDSTR progname = "MSORT";
29: PACKEDSTR usagemes =
30: "機 能:テキストをソートします¥n¥
31: 使用法:%s [スイッチ] [入力ファイル[…]] [/O出力ファイル]¥n¥
32: ¥t/D¥tDUMPDIC.X形式の単語ファイルを見出し語についてソートする
¥n¥
33: ¥t/I¥t半角英大文字と小文字を区別しない¥n¥
34: ¥t/Ln¥t一度に処理する最大行数の指定(無指定時は5000行)¥n¥
35: ¥t/Kn¥t行のn文字目以降を比較対象にする¥n¥
36: ¥t/M¥tすでにソート済みの複数ファイルを併合する¥n¥
37: ¥t/R¥t降順にソートする¥n¥
38: ¥t/U¥t重複行を1度しか出力しない¥n¥
39: ¥t/V¥t実行経過を報告する¥n";
40:
41: static PACKEDSTR GHOSTMES =
42: "ファイル[%s]がみつかりません(さっきまではあったのに!)";
43:
44: typedef struct sortdata {
45:     STRPTR  key;
46:     STRPTR  lin;
47:     FLPTR   flp;
48: } DATA, *DATAPTR;
49:
50: static DATAPTR lintable;
51:
```


▶我が家のX68000はタバコのヤニで色が世界で1台の黄灰色になっています。タバコやめようかな……。
武藤　俊哉 (24) 神奈川県

```
 52: static STRPTR linbuf;
 53: static FILE *destfp;
 54: static int ifilectr = 0;
 55: static int ofilectr = 0;
 56: static int nline = 0;
 57: static int keypos = 0;
 58: static int (*cmpfunc)();
 59:
 60: static boolean dicmode = FALSE;
 61: static boolean ignorecase = FALSE;
 62: static boolean mergeonly = FALSE;
 63: static boolean revmode = FALSE;
 64: static boolean uniq = FALSE;
 65: static boolean verbose = FALSE;
 66:
 67: static void sort0( left, right )
 68: DATAPTR left, right;
 69: {
 70:     DATAPTR ll, rr;
 71:     DATA temp;
 72:     STRPTR mm;
 73:
 74:     mm = ( left + ( right - left ) / 2 )->key;
 75:
 76:     for ( ll = left - 1, rr = right + 1; ; ) {
 77:         for ( ; (*cmpfunc)( mm, (++ll)->key ) > 0; );
 78:         for ( ; (*cmpfunc)( (--rr)->key, mm ) > 0; );
 79:         if ( ll >= rr )
 80:             break;
 81:         temp = *ll; *ll = *rr; *rr = temp;
 82:     }
 83:
 84:     if ( ++rr < right )
 85:         sort0( rr, right );
 86:     if ( left < --ll )
 87:         sort0( left, ll );
 88: }
 89:
 90: static void sort( dataary, n )
 91: DATAPTR dataary;
 92: int n;
 93: {
 94:     if ( n <= 1 )
 95:         return;
 96:
 97:     sort0( dataary, dataary + n - 1 );
 98: }
 99:
100: static void s_sort( dataary, n )
101: DATAPTR dataary;
102: int n;
103: {
104:     DATAPTR p, q;
105:     DATA temp;
106:     STRPTR mm;
107:     int gap;
108:
109:     for ( gap = n / 2; gap > 0; gap /= 2 ) {
110:         for ( p = dataary + gap; p < dataary + n; p++ ) {
111:             temp = *p;
112:             mm = p->key;
113:             for ( q = p - gap;
114:                 q >= dataary && (*cmpfunc)( q->key, mm ) > 0;
115:                 q[ gap ] = *q, q -= gap );
116:             q[ gap ] = temp;
117:         }
118:     }
119: }
120:
121: static STRPTR getindex( str )
122: STRPTR str;
123: {
124:     STRPTR temp, p;
125:
126:     if ( dicmode ) {
127:         for ( p = temp = memalloc( strlen( str ) * 2 + 2 ); *
str > '¥x20'; )
128:             *p++ = *str++;
129:         *p = '¥0';
130:         SJIStoASK( temp, strzenkata( temp, temp ) );
131:         p = realloc( temp, strlen( temp ) + 1 );
132:     } else if ( ignorecase ) {
133:         strtoupper( p = dupstr( str ) );
134:     } else {
135:         p = str;
136:     }
137:     return ( p );
138: }
139:
140: static int mystrcmp( s1, s2 )
141: STRPTR s1, s2;
142: {
143:     s1 = ( strlen( s1 ) < keypos ) ? "" : s1 + keypos;
144:     s2 = ( strlen( s2 ) < keypos ) ? "" : s2 + keypos;
145:     return ( strcmp( s1, s2 ) );
146: }
147:
148: static int myrstrcmp( s1, s2 )
149: STRPTR s1, s2;
150: {
151:     return( -mystrcmp( s1, s2 ) );
152: }
153:
154: static FLPTR getsflp;
155:
156: static int getstr()
157: {
158:     FLPTR flp;
159:     boolean done;
160:
161:     KEYSNS();
162:     for ( done = FALSE; !done; ) {
163:         if ( ( flp = getsflp ) == NULL )
164:             return ( EOF );
165:         if ( flp->fp == NULL ) {
166:             if ( flp->name == NULL ) {
167:                 flp->fp = stdin;
168:                 if ( verbose )
169:                     fputs( "[標準入力]¥n", stderr );
170:             } else {
171:                 memtest( BUFSIZ );
172:                 if ( flpopen( flp, "r" ) == NULL )
173:                     fatal_error( GHOSTMES, flp->name );
174:                 if ( verbose )
175:                     fprintf( stderr, "%s¥n", flp->name );
176:             }
177:         }
178:         done = ( fgets( linbuf, LINBUFSIZ, flp->fp ) != NULL
);
179:         if ( feof( flp->fp ) ) {
180:             getsflp = flpclose( flp );
181:             free( flp );
182:             ifilectr--;
183:         }
184:     }
185:     return ( !EOF );
186: }
187:
188: static void setlline( p )
189: DATAPTR p;
190: {
191:     p->lin = dupstr( linbuf );
192:     p->key = getindex( p->lin );
193: }
194:
195: static int readfile()
196: {
197:     STRPTR temp;
198:     DATAPTR p;
199:     int linctr;
200:
201:     if ( verbose )
202:         fputs( "テキストを読み込みます¥n", stderr );
203:
204:     for ( p = lintable, linctr = 0;
205:         linctr < nline && getstr() != EOF; linctr++, p++ ) {
206:         KEYSNS();
207:         setlline( p );
208:         free( temp = malloc( RSV ) );
209:         if ( temp == NULL )
210:             break;
211:     }
212:     return ( linctr );
213: }
214:
215: static FLPTR tempflp;
216:
217: static FILE *createtemp()
218: {
219:     FLPTR flp;
220:     static STRPTR tempname = NULL;
221:     STRPTR p;
222:
223:     if ( tempname == NULL ) {
224:         tempname = memalloc( NCHAR * 2 );
225:         if ( ( p = getenv( "temp" ) ) == NULL )
226:             p = "";
227:         sprintf( tempname, "%s$1234567", p );
228:         tempname = realloc( tempname, strlen( tempname ) + 1
);
229:     }
230:     if ( ofilectr >= NTEMP )
231:         fatal_error( TOPENERRMES );
232:     sprintf( strchr( tempname, '¥0' ) - 7, "%.7X", ofilectr++
);
233:     tempflp = addtofilelist( tempname );
234:     memtest( BUFSIZ );
235:     if ( flpopen( tempflp, "w" ) == NULL )
236:         fatal_error( TOPENERRMES );
237:     tempflp->istemp = TRUE;
238:     return ( tempflp->fp );
239: }
240:
241: static void killtemps()
242: {
243:     FLPTR flp;
244:
245:     for ( flp = getfilelist(); ofilectr > 0 && flp != NULL;
246:             ofilectr--, flp = flp->next )
247:         killtemp( flp );
248: }
249:
250: static void writefile( linctr )
251: int linctr;
252: {
253:     DATAPTR p;
254:     STRPTR lstlin;
255:     FILE *fp;
256:
257:     if ( ifilectr + ofilectr == 0 ) {
258:         fp = destfp;
259:         if ( verbose )
260:             fputs( "テキストを書き出します¥n", stderr );
261:     } else {
262:         fp = createtemp();
263:         if ( verbose )
264:             fputs( "テキストを作業用ファイルに書き出します¥n"
, stderr );
265:     }
266:
```

```
267:     lstlin = dupstr( "" );
268:     for ( p = lintable; linctr--; p++ ) {
269:         KEYSNS();
270:         if ( uniq && !strcmp( lstlin, p->lin ) ) {
271:             if ( p->lin != p->key )
272:                 free( p->lin );
273:         } else {
274:             free( lstlin );
275:             if ( fputs( p->lin, fp ) == EOF )
276:                 diskfull( fp );
277:             lstlin = p->lin;
278:         }
279:         if ( p->key != lstlin )
280:             free( p->key );
281:     }
282:     free( lstlin );
283:     fclose( fp );
284:     if ( fp != destfp )
285:         tempflp->fp = NULL;
286: }
287:
288: static void mergefile()
289: {
290:     FILE *fp;
291:     FLPTR flp;
292:     DATAPTR p;
293:     STRPTR lstlin;
294:     int filectr, fileleft;
295:
296:     for ( flp = getfilelist(), fileleft = ifilectr; fileleft
 > 0; ) {
297:         if ( verbose )
298:             fputs( "マージします¥n", stderr );
299:         for ( filectr = 0, p = lintable;
300:             flp != NULL && fileleft > 0 && filectr < NMERGE;
301:                                 flp = flp->next, fileleft-- )
 {
302:             memtest( BUFSIZ );
303:             if ( flpopen( flp, "r" ) == NULL )
304:                 fatal_error( GHOSTMES, flp->name );
305:             if ( fgets( linbuf, LINBUFSIZ, flp->fp ) != NULL
 ) {
306:                 p->flp = flp;
307:                 setlline( p );
308:                 if ( verbose )
309:                     fprintf( stderr, "%s¥n",
310:                     ( flp->name != NULL ) ? flp->name : "[標
準入力]" );
311:                 filectr++;
312:                 p++;
313:             } else {
314:                 killtemp( flp );
315:             }
316:         }
317:
318:         if ( fileleft > 0 ) {
319:             fp = createtemp();
320:             fileleft++;
321:         } else {
322:             fp = destfp;
323:         }
324:
325:         lstlin = dupstr( "" );
326:         for ( p = lintable; filectr > 0; ) {
327:             KEYSNS();
328:             if ( filectr > 1 )
329:                 sort( p, filectr );
330:             if ( uniq && !strcmp( lstlin, p->lin ) ) {
331:                 if ( p->lin != p->key )
332:                     free( p->lin );
333:             } else {
334:                 free( lstlin );
335:                 if ( fputs( p->lin, fp ) == EOF )
336:                     diskfull( fp );
337:                 lstlin = p->lin;
338:             }
339:             if ( p->key != lstlin )
340:                 free( p->key );
341:             if ( fgets( linbuf, LINBUFSIZ, p->flp->fp ) != NU
LL ) {
342:                 setlline( p );
343:             } else {
344:                 killtemp( p->flp );
345:                 p++;
346:                 filectr--;
347:             }
348:         }
349:         free( lstlin );
350:         fclose( fp );
351:     }
352: }
353:
354: void main( argc, argv )
355: int argc;
356: STRPTR *argv;
357: {
358:     boolean dflt = TRUE;
359:     STRPTR destfile = NULL;
360:     int linctr = 0;
361:
362:     for ( ; --argc; ) {
363:         if ( strchr( SWITCH_DELIMITER, **++argv ) != NULL ) {
364:             unsigned int sc;
365:             sc = *++*argv;
366:             switch ( tolower( sc ) ) {
367:                 case '¥0':
368:                     dflt = FALSE;
369:                     addtofilelist( NULL );
370:                     ifilectr++;
371:                     break;
372:                 case 'd':
373:                     setflag( &dicmode );
374:                     break;
375:                 case 'i':
376:                     setflag( &ignorecase );
377:                     break;
378:                 case 'k':
379:                     setvalue( &keypos, ++*argv, 1, INT_MAX );
380:                     break;
381:                 case 'l':
382:                     setvalue( &nline, ++*argv, 100, INT_MAX )
;
383:                     break;
384:                 case 'm':
385:                     setflag( &mergeonly );
386:                     break;
387:                 case 'o':
388:                     setptr( &destfile, ++*argv );
389:                     break;
390:                 case 'r':
391:                     setflag( &revmode );
392:                     break;
393:                 case 'u':
394:                     setflag( &uniq );
395:                     break;
396:                 case 'v':
397:                     setflag( &verbose );
398:                     break;
399:                 default:
400:                     usage();
401:                     break;
402:             }
403:         } else {
404:             dflt = FALSE;
405:             ifilectr += setfilename( *argv );
406:         }
407:     }
408:
409:     if ( dflt ) {
410:         addtofilelist( NULL );
411:         ifilectr++;
412:     }
413:     if ( ifilectr == 0 )
414:         fatal_error( "指定のファイルは存在しません" );
415:
416:     allmem();
417:     keypos = ( !dicmode && keypos > 0 ) ? keypos - 1 : 0;
418:     uniq = ( dicmode ) ? FALSE : uniq;
419:     nline = ( nline == 0 ) ? NLINE : nline;
420:     cmpfunc = ( revmode ) ? myrstrcmp : mystrcmp;
421:
422:     destfp = setstdout( destfile );
423:     resetstdin();
424:     breakset( killtemps );
425:
426:     linbuf = memalloc( LINBUFSIZ );
427:     lintable = memalloc( nline * sizeof( *lintable ) );
428:
429:     if ( !mergeonly ) {
430:         for ( getsflp = getfilelist(), newfilelist(); ifilect
r > 0; ) {
431:             if ( ( linctr = readfile() ) == 0 )
432:                 break;
433:             if ( verbose )
434:                 fprintf( stderr, "テキスト(%d行)をソートしま
す¥n", linctr );
435:             sort( lintable, linctr );
436:             writefile( linctr );
437:         }
438:         ifilectr = ofilectr;
439:     }
440:     if ( ifilectr > 0 )
441:         mergefile();
442:
443:     if ( verbose )
444:         fputs( "完了しました¥n", stderr );
445:
446:     exit( EXIT_SUCCESS );
447: }
```

リスト6 wild.c

```
 1: #include    "mydef.h"
 2: #include    <stdio.h>
 3: #include    <stdlib.h>
 4: #include    <string.h>
 5: #include    <io.h>
 6: #include    <doslib.h>
 7: #include    "wild.h"
 8: #include    "misc.h"
 9: #include    "myerror.h"
10:
11: static FLPTR filelisttop = NULL;
12:
13: FLPTR getfilelist()
14: {
15:     return ( filelisttop );
16: }
17:
18: void newfilelist()
```


▶アンケートはがきの「あなたの愛機は」でX68000の無印がいつのまにか初代になってんじゃん。前のほうがよかったな。　　田中　進之祐 (16) 東京都

```
 19: {
 20:     filelisttop = NULL;
 21: }
 22:
 23: FLPTR addtofilelist( name )
 24: STRPTR name;
 25: {
 26:     static FLPTR flp;
 27:     FLPTR p;
 28:
 29:     p = memalloc( sizeof( FILELIST ) );
 30:     if ( filelisttop == NULL )
 31:         filelisttop = p;
 32:     else
 33:         flp->next = p;
 34:     flp = p;
 35:     flp->next = NULL;
 36:     flp->fp = NULL;
 37:     flp->name = ( name != NULL ) ? dupstr( name ) : NULL;
 38:     flp->istemp = FALSE;
 39:
 40:     return ( flp );
 41: }
 42:
 43: FILE *flpopen( flp, mode )
 44: FLPTR flp;
 45: STRPTR mode;
 46: {
 47:     if ( flp->name == NULL )
 48:         return ( flp->fp = stdin );
 49:     else
 50:         return ( flp->fp = fopen( flp->name, mode ) );
 51: }
 52:
 53: FLPTR flpclose( flp )
 54: FLPTR flp;
 55: {
 56:     if ( flp->fp != NULL )
 57:         fclose( flp->fp );
 58:     flp->fp = NULL;
 59:     free( flp->name );
 60:     return ( flp->next );
 61: }
 62:
 63: void killtemp( flp )
 64: FLPTR flp;
 65: {
 66:     STRPTR name;
 67:
 68:     name = flp->name;
 69:     flpclose( flp );
 70:     if ( flp->istemp && name != NULL )
 71:         unlink( name );
 72: }
 73:
 74: static int wild( buff )
 75: STRPTR buff;
 76: {
 77:     struct  FILBUF filbuf;
 78:     struct  NAMECKBUF nambuf;
 79:     int fstat;
 80:     FILE *fp;
 81:     int ctr = 0;
 82:
 83:     NAMECK( buff, &nambuf );
 84:     for ( fstat = FILES( &filbuf, buff, 0x20 ); fstat == 0;
 85:                                          fstat = NFILES( &filb
uf ) ) {
 86:         sprintf( buff, "%s%s", nambuf.drive, filbuf.name );
 87:         addtofilelist( buff );
 88:         ctr++;
 89:     }
 90:     return ( ctr );
 91: }
 92:
 93: int setfilename( name )
 94: STRPTR name;
 95: {
 96:     struct  NAMECKBUF nambuf;
 97:     struct  FILBUF filbuf;
 98:     int nameckstat;
 99:     STR temp;
100:     FILE *fp;
101:     int ctr = 0;
102:
103:     if ( ( nameckstat = NAMECK( name, &nambuf ) ) < 0 ) {
104:         fatal_error( ILLNAMMES, name );
105:     } else {
106:         sprintf( temp, "%s%s%s", nambuf.drive, nambuf.name, n
ambuf.ext );
107:         if ( nameckstat == 0 && FILES( &filbuf, temp, 0x10 )
== 0 )
108:             strcat( temp, "¥¥*.*" );
109:         else if ( nameckstat == 0xff )
110:             strcat( temp, "*.*" );
111:         ctr += wild( temp );
112:     }
113:     if ( ctr == 0 && nameckstat == 0 ) {
114:         if ( ( fp = fopen( name, "r" ) ) != NULL ) {
115:             if ( isatty( fileno( fp ) ) ) {
116:                 addtofilelist( name );
117:                 ctr++;
118:             }
119:             fclose( fp );
120:         } else {
121:             fatal_error( ROPENERRMES, name );
122:         }
123:     }
124:     return ( ctr );
125: }
```

リスト7 wild.h

```
 1: typedef struct _filelist {
 2:     struct _filelist *next;
 3:     FILE     *fp;
 4:     STRPTR   name;
 5:     short    istemp;
 6: } FILELIST, *FLPTR;
 7:
 8: FLPTR    getfilelist( void );
 9: void     newfilelist( void );
10: FLPTR    addtofilelist( STRPTR );
11: FILE     *flpopen( FLPTR, STRPTR );
12: FLPTR    flpclose( FLPTR );
13: void     killtemp( FLPTR );
14: int      setfilename( STRPTR );
```

リスト8 (1000バイトでセーブ)

```
0000  60 00 01 0C 2F 43 48 44  : 6B
0008  49 43 2F 00 00 00 6A 00  : 25
0010  00 00 00 00 00 00 00 02  : 02
0018  4E F9 00 00 00 00 4E F9  : 8E
0020  00 00 00 00 30 3A 00 E2  : 4C
0028  6B EE 67 14 48 E7 00 84  : 87
0030  61 62 2B 48 00 0A 53 6D  : 00
0038  00 08 4C DF 21 00 4E 75  : 17
0040  61 D6 61 38 66 34 61 5C  : 27
0048  60 DA 30 3A 00 BC 6B CE  : 99
0050  67 18 48 E7 00 84 61 3C  : CF
0058  66 0A 2B 48 00 0A 53 6D  : AD
0060  00 08 70 00 4C DF 21 00  : C4
0068  4E 75 61 B2 4A 80 67 0A  : 11
0070  61 0A 66 06 61 A2 61 2C  : 67
0078  70 00 4E 75 0C 40 59 00  : D8
-----------------------------------
SUM:  70 ED 97 15 31 2D 63 90  6666

0080  66 10 70 02 4E 4F 02 40  : C7
0088  00 0F 67 06 20 3C 00 01  : D9
0090  59 00 4E 75 4B FA 00 6A  : CB
0098  70 01 48 40 20 6D 00 0A  : 90
00A0  30 18 4E 75 48 E7 C0 C4  : BE
00A8  4B FA 00 56 2F 3A FF 66  : 69
00B0  2F 3C 00 00 00 01 FF 22  : 8D
00B8  50 8F 00 7C 07 00 3B 7C  : 19
00C0  00 05 00 08 41 ED 00 0E  : 49
00C8  2B 48 00 0A 43 FA FF 3E  : F7
00D0  20 D9 20 D9 22 6D 00 00  : 81
00D8  20 11 66 04 20 2D 00 04  : EC
00E0  22 40 2B 40 00 00 58 89  : AE
00E8  70 00 10 19 67 08 30 C0  : F8
00F0  52 6D 00 08 60 F4 30 FC  : 47
00F8  1D 0D 4C DF 23 03 4E 75  : 3E
-----------------------------------
SUM:  95 EE C8 33 07 94 00 87  5B08

0100  00 00 00 00 00 00 00 00  : 00
0108  FF FF 00 00 00 00 48 7A  : C0
0110  01 F8 FF 09 49 E8 01 00  : 33
0118  2A 49 41 E8 00 10 43 E9  : D8
0120  24 3E 93 C8 48 E7 00 C0  : AC
0128  FF 4A 4A 80 6B 00 00 9E  : 1C
0130  4F ED 24 3E 61 00 01 02  : 02
0138  61 00 00 A4 4A 05 67 0C  : C7
0140  61 00 00 D0 48 7A 01 EA  : DE
0148  FF 09 FF 00 4A 47 66 76  : 74
0150  4A 46 67 66 32 3C 01 00  : CC
0158  43 FA FE CA 70 80 4E 4F  : 92
0160  29 40 00 1A 32 3C 01 01  : F3
0168  43 FA FE E0 70 80 4E 4F  : A8
0170  29 40 00 20 45 FA FF 8A  : 51
0178  43 FA 00 64 24 89 25 49  : BC
-----------------------------------
SUM:  C2 72 A3 99 E6 A0 1D A1  CC31

0180  00 04 41 ED 00 00 2C 49  : A7
0188  58 89 20 08 12 D8 66 FC  : 55
0190  53 89 20 40 20 09 52 80  : 37
0198  08 80 00 00 22 40 2C 89  : 9F
01A0  41 E8 00 5C 4A 10 66 DE  : 23
01A8  42 96 41 FA FF 5C 42 50  : 00
01B0  93 CC 52 89 42 67 2F 09  : 1B
01B8  FF 31 41 FA 01 DF 60 10  : BB
01C0  41 FA 01 B7 60 0A 41 FA  : 98
01C8  01 91 60 04 41 FA 01 7A  : AC
01D0  3F 3C 00 02 2F 08 FF 1E  : D1
01D8  3F 3C 00 01 FF 4C 42 A7  : B0
01E0  FF 20 2E 80 32 3C 01 00  : 3C
01E8  93 C9 70 80 4E 4F 22 40  : 4B
01F0  26 40 70 80 4E 4F 7E 00  : 71
01F8  20 3A FE 0A B0 AB FF E0  : 9C
-----------------------------------
SUM:  60 77 C2 56 2D B0 6A EE  BC61

0200  66 0A 20 3A FE 04 B0 AB  : 27
0208  FF E4 57 C7 FF 20 58 8F  : 07
0210  4E 75 4A 47 67 AA 32 3C  : D3
0218  01 00 22 6B FF F6 70 80  : 73
0220  4E 4F 32 3C 01 01 22 6B  : 9A
0228  FF FC 70 80 4E 4F 48 6B  : 3B
0230  FE EC FF 49 58 8F 4E 75  : DC
0238  4A 1A 67 00 FF 7E 7C 00  : C4
0240  7A 00 41 ED 02 E2 43 ED  : BC
0248  00 00 74 07 61 5E 4A 12  : 96
0250  67 26 61 7C 48 6D 03 E2  : 04
0258  2F 08 FF 37 50 8F 4A 80  : 16
0260  66 00 FF 58 61 16 43 E9  : 60
0268  00 5C 52 46 51 CA FF DE  : EC
0270  61 3A 4A 12 66 00 FF 44  : A0
0278  42 11 4E 75 48 E7 00 C0  : 05
-----------------------------------
SUM:  62 89 E9 84 64 24 F9 6D  3A88

0280  41 ED 03 E2 12 D8 66 FC  : 5F
0288  53 89 41 ED 04 25 12 D8  : 1D
0290  66 FC 53 89 41 ED 04 38  : A8
0298  4A 10 66 04 41 FA 01 45  : 45
02A0  12 D8 66 FC 53 89 4C DF  : 53
02A8  03 00 4E 75 61 4C 0C 12  : 91
02B0  00 2F 67 08 0C 12 00 2D  : E9
02B8  67 02 4E 75 52 8A 10 1A  : 32
02C0  08 80 00 05 0C 00 00 52  : EB
02C8  66 00 FE F0 7A FF 60 DC  : 09
02D0  2F 08 4A 12 67 1C 0C 12  : 34
02D8  00 20 67 16 0C 12 00 09  : C4
02E0  67 10 0C 12 00 2D 67 0A  : 33
02E8  0C 12 00 2F 67 04 10 DA  : A2
02F0  60 E0 42 10 20 5F 4E 75  : D4
02F8  52 8A 0C 12 00 20 67 F8  : 79
-----------------------------------
SUM:  82 BF 6F CA 2A 32 7D 23  22B2

0300  0C 12 00 09 67 F2 4E 75  : 43
0308  41 53 4B 36 38 4B 20 44  : FC
0310  69 63 74 69 6F 6E 61 72  : 59
0318  79 20 46 69 6C 65 20 53  : 8C
0320  65 6C 65 63 74 6F 72 20  : 0E
0328  76 31 2E 30 30 0D 0A 00  : 4C
0330  43 48 44 49 43 82 F0 90  : 5D
0338  D8 82 E8 95 FA 82 B5 82  : 8A
0340  DC 82 B5 82 BD 0D 0A 00  : 69
```

▶特集の吉田幸一さんへ。私はこれからのさくぶんの時間には，あなたの記事を思い出し，よい文を書くように努力いたします。これからもよい記事書いてくださいね。

辻村　貴志 (17) 北海道

```
0348  83 81 83 82 83 8A 95 73  : 1E
0350  91 AB 82 C5 82 B7 0D 0A  : D3
0358  00 43 48 44 49 43 82 CD  : AA
0360  82 B7 82 C5 82 C9 91 67  : C3
0368  82 DD 8D 9E 82 DD 8D CF  : 45
0370  82 DD 82 C5 82 B7 0D 0A  : F6
0378  00 43 48 44 49 43 82 CD  : AA
----------------------------------
SUM:  9B F4 9F FB 35 C1 EB 07  1ACE
```

```
0380  82 DC 82 BE 91 67 82 DD  : F5
0388  8D 9E 82 DC 82 EA 82 C4  : 3B
0390  82 A2 82 DC 82 B9 82 F1  : 30
0398  0D 0A 00 8E 67 97 70 96  : A9
03A0  40 81 46 43 48 44 49 43  : 62
03A8  20 5B 83 58 83 43 83 62  : 01
03B0  83 60 5D 20 8E AB 8F 91  : B9
03B8  83 74 83 40 83 43 83 8B  : 8E
03C0  20 5B 81 63 5D 0D 0A 09  : DC
```

```
03C8  2F 52 09 43 48 44 49 43  : E5
03D0  82 CC 8F ED 92 93 82 F0  : 61
03D8  89 F0 8F 9C 82 B7 82 E9  : 48
03E0  0D 0A 00 2E 44 49 43 00  : 15
03E8  00 00 00 00 00 00 00 00  : 00
03F0  00 00 00 00 00 00 00 00  : 00
03F8  00 00 00 00 00 00 00 00  : 00
----------------------------------
SUM:  6B 49 D7 5C D5 FA 6E 0E  E560
```

リスト9 chdic.s(参考)

```
  1:          .list
  2: *****************************************************
  3: *    ASK68K Dictionary File Selector v1.00          *
  4: *****************************************************
  5:          .include        iocscall.mac
  6:          .include        doscall.mac
  7:          .include        files.h
  8:          .include        const.h
  9: *
 10: DOSLIKE equu    -1
 11: *
 12: HOTKEY  equ     $5900           *XF5
 13: CRKEY   equ     $1d0d
 14: MAXDIC  equ     8
 15: *
 16: STRCPY  macro
 17:         local   loop
 18: loop:   move.b  (a0)+,(a1)+
 19:         bne     loop
 20:         subq.l  #1,a1
 21:         endm
 22: *
 23:         .offset 0
 24: *
 25: _PROGST:        .ds.w   2
 26: _ID:            .ds.w   4
 27:                 .ds.w   6
 28:                 .ds.w   1
 29: _IOCS00:        .ds.l   1
 30:                 .ds.w   1
 31: _IOCS01:        .ds.l   1
 32: BENT:
 33: *
 34:         .offset 0
 35: *
 36: DICTBL:         .ds.b   NAMBUFSIZ*MAXDIC
 37:                 .ds.w   1
 38: TEMP:           .ds.b   256
 39: NAMBUF:         .ds.b   NAMBUFSIZ
 40:                 .ds.l   2048
 41: INISP:
 42: WORKSIZE:
 43: *
 44:         .offset 0
 45: *
 46: CURDIC:         .ds.l   1
 47: FSTDIC:         .ds.l   1
 48: KCNT:           .ds.w   1
 49: KPTR:           .ds.l   1
 50: KBUF:
 51: *
 52:         .text
 53:         .even
 54: *
 55: RFLAGREG        equ     d5
 56: DICCNTREG       equ     d6
 57: KPFLAGREG       equ     d7
 58:
 59: KPPROG          equ     a3
 60: PROGBASE        equ     a4
 61: WORK            equ     a5
 62: *
 63: progst:
 64:         bra.w   init
 65: ID:     .dc.b   '/CHDIC/',0
 66: dickey: .dc.w   $0000,$6a00,$0000,0     *F8
 67: askmod: .dc.l   2                       *一括変換(先読みあり)
 68: IOCS00: jmp     0
 69: IOCS01: jmp     0
 70: *
 71: KEYINP:
 72:         move.w  kcnt(pc),d0
 73:         bmi     IOCS00
 74:         beq     kinp1
 75:
 76:         movem.l a0/WORK,-(sp)
 77:         bsr     sns
 78:         move.l  a0,KPTR(WORK)
 79:         subq.w  #1,KCNT(WORK)
 80:
 81: kinp0:  movem.l (sp)+,a0/WORK
 82:         rts
 83: *
 84: kinp1:  bsr     IOCS00
 85:         bsr     keychk
 86:         bne     retn
 87:         bsr     do
 88:         bra     KEYINP
 89:
 90: *
 91: KEYSNS:
 92:         move.w  kcnt(pc),d0
 93:         bmi     IOCS01
 94:         beq     ksns1
 95:
 96:         movem.l a0/WORK,-(sp)
 97:         bsr     sns
 98:         bne     ksns0
 99:         move.l  a0,KPTR(WORK)
100:         subq.w  #1,KCNT(WORK)
101:         moveq.l #0,d0
102: ksns0:  movem.l (sp)+,a0/WORK
103:         rts
104:
105: ksns1:  bsr     IOCS01
106:         tst.l   d0
107:         beq     retn
108:         bsr     keychk
109:         bne     retn
110:         bsr     IOCS00
111:         bsr     do
112:         moveq.l #0,d0
113: retn:   rts
114: *
115: keychk:
116:         cmpi.w  #HOTKEY,d0
117:         bne     kchk0
118:         IOCS    _B_SFTSNS
119:         andi.w  #$0f,d0
120:         beq     kchk0
121:         move.l  #$0001_0000|HOTKEY,d0
122: kchk0:  rts
123: *
124: sns:
125:         lea.l   work(pc),WORK
126:         moveq.l #1,d0
127:         swap.w  d0
128:         movea.l KPTR(WORK),a0
129:         move.w  (a0)+,d0
130:         rts
131: *
132: do:
133:         movem.l d0-d1/a0-a1/WORK,-(sp)
134:
135:         lea.l   work(pc),WORK
136:
137:         move.l  askmod(pc),-(sp)
138:         move.l  #1,-(sp)
139:         DOS     _KNJCTRL
140:         addq.l  #8,sp
141:
142: do0:    ori.w   #$0700,sr
143:
144:         move.w  #5,KCNT(WORK)
145:         lea.l   KBUF(WORK),a0
146:         move.l  a0,KPTR(WORK)
147:         lea.l   dickey(pc),a1
148:         move.l  (a1)+,(a0)+
149:         move.l  (a1)+,(a0)+
150:
151:         movea.l CURDIC(WORK),a1
152:         move.l  (a1),d0
153:         bne     do1
154:         move.l  FSTDIC(WORK),d0
155: do1:    movea.l d0,a1
156:         move.l  d0,CURDIC(WORK)
157:         addq.l  #4,a1
158:
159:         moveq.l #0,d0
160: do2:    move.b  (a1)+,d0
161:         beq     do3
162:         move.w  d0,(a0)+
163:         addq.w  #1,KCNT(WORK)
164:         bra     do2
165:
166: do3:    move.w  #CRKEY,(a0)+
167:
168: do9:    movem.l (sp)+,d0-d1/a0-a1/WORK
169:         rts
170:
171: *
172: work:
173: curdic: .dc.l   0
174: fstdic: .dc.l   0
175: kcnt:   .dc.w   -1      *uninitialized
176: kptr:   .dc.l   0
177: kbuf:
178: *
179: init:
180:         pea.l   ttlmes(pc)
181:         DOS     _PRINT
182:
183:         lea.l   $0100(a0),PROGBASE
184:         movea.l a1,WORK
185:
186:         lea.l   16(a0),a0
187:         lea.l   WORKSIZE(a1),a1
188:         suba.l  a0,a1
189:         movem.l a0-a1,-(sp)
190:         DOS     _SETBLOCK
191:         tst.l   d0
192:         bmi     error0
193:
194:         lea.l   INISP(WORK),sp
195:
196:         bsr     chkarg
197:
198:         bsr     chkkp
```


▶9月いっぱいまで夏休みだというのに，X68000をいじるひまもない。運動部には夏休みもないのである。ああ，海に行きたい……。P.S.ユニフォームのせいで土方焼けしてしまった。
小林　勝（22）千葉県

```
199:            tst.b   RFLAGREG
200:            beq     keep
201: *
202: koff:
203:            bsr     kpoff
204:
205:            pea.l   mes1(pc)
206:            DOS     _PRINT
207:
208:            DOS     _EXIT
209: keep:
210:            tst.w   KPFLAGREG
211:            bne     error1
212:
213:            tst.w   DICCNTREG
214:            beq     usage
215:            move.w  #$0100,d1
216:            lea.l   KEYINP(pc),a1
217:            IOCS    _B_INTVCS
218:            move.l  d0,_IOCS00(PROGBASE)
219:
220:            move.w  #$0101,d1
221:            lea.l   KEYSNS(pc),a1
222:            IOCS    _B_INTVCS
223:            move.l  d0,_IOCS01(PROGBASE)
224:
225:            lea.l   curdic(pc),a2
226:            lea.l   dictbl(pc),a1
227:            move.l  a1,(a2)
228:            move.l  a1,FSTDIC-CURDIC(a2)
229:            lea.l   DICTBL(WORK),a0
230: keep0:     movea.l a1,a6
231:            addq.l  #4,a1
232:            move.l  a0,d0
233:            STRCPY
234:            movea.l d0,a0
235:
236:            move.l  a1,d0
237:            addq.l  #1,d0
238:            bclr.l  #0,d0
239:            movea.l d0,a1
240:
241:            move.l  a1,(a6)
242:            lea.l   NAMBUFSIZ(a0),a0
243:            tst.b   (a0)
244:            bne     keep0
245:            clr.l   (a6)
246:            lea.l   kcnt(pc),a0
247:            clr.w   (a0)
248:            suba.l  PROGBASE,a1
249:            addq.l  #1,a1
250:            clr.w   -(sp)
251:            move.l  a1,-(sp)
252:            DOS     _KEEPPR
253: *
254: usage:
255:            lea     usgmes(pc),a0
256:            bra     errret
257: *
258: error2:    lea.l   errms2(pc),a0
259:            bra     errret
260: error1:    lea.l   errms1(pc),a0
261:            bra     errret
262: error0:    lea.l   errms0(pc),a0
263: errret:    move.w  #STDERR,-(sp)
264:            move.l  a0,-(sp)
265:            DOS     _FPUTS
266:            move.w  #1,-(sp)
267:            DOS     _EXIT2
268: dictbl:
269:
270: KBUFSIZ equ        dictbl-kbuf          *KBUFSIZ>200 !
271:
272: ************************************************************
273:
274: chkkp:
275:            clr.l   -(sp)
276:            DOS     _SUPER
277:            move.l  d0,(sp)
278:
279:            move.w  #$0100|_B_KEYINP,d1
280:            suba.l  a1,a1
281:            IOCS    _B_INTVCS
282:            movea.l d0,a1
283:            movea.l d0,KPPROG
284:            IOCS    _B_INTVCS
285:
286:            moveq.l #0,KPFLAGREG
287:            move.l  ID(pc),d0
288:            cmp.l   _ID-BENT(KPPROG),d0
289:            bne     chkkp0
290:            move.l  ID+4(pc),d0
291:            cmp.l   _ID+4-BENT(KPPROG),d0
292:            seq.b   KPFLAGREG
293:
294: chkkp0:    DOS     _SUPER
295:            addq.l  #4,sp
296:            rts
297: *
298: kpoff:
299:            tst.w   KPFLAGREG
300:            beq     error2
301:            move.w  #$0100|_B_KEYINP,d1
302:            movea.l _IOCS00-BENT(KPPROG),a1
303:            IOCS    _B_INTVCS
304:            move.w  #$0100|_B_KEYSNS,d1
305:            movea.l _IOCS01-BENT(KPPROG),a1
306:            IOCS    _B_INTVCS
307:            pea.l   -256+16-BENT(KPPROG)
308:            DOS     _MFREE
309:            addq.l  #4,sp
310:            rts
311: *
312: chkarg:
313:            tst.b   (a2)+
314:            beq     usage
315:
316:            moveq.l #0,DICCNTREG
317:            moveq.l #0,RFLAGREG
318:
319:            lea.l   TEMP(WORK),a0
320:            lea.l   DICTBL(WORK),a1
321:            moveq.l #MAXDIC-1,d2
322: ckarg0:    bsr     nextarg
323:
324:            tst.b   (a2)
325:            beq     ckarg1
326:
327:            bsr     getarg
328:
329:            pea.l   NAMBUF(WORK)
330:            move.l  a0,-(sp)
331:            DOS     _NAMECK
332:            addq.l  #8,sp
333:            tst.l   d0
334:            bne     usage
335:            bsr     setfnam
336:            lea.l   NAMBUFSIZ(a1),a1
337:            addq.w  #1,DICCNTREG
338:
339:            dbra    d2,ckarg0
340:
341:            bsr     nextarg
342:            tst.b   (a2)
343:            bne     usage
344:
345: ckarg1:    clr.b   (a1)
346:            rts
347: *
348: setfnam:
349:            movem.l a0-a1,-(sp)
350:            lea.l   NAMBUF+DRIVE(WORK),a0
351:            STRCPY
352:            lea.l   NAMBUF+NAME(WORK),a0
353:            STRCPY
354:            lea.l   NAMBUF+EXT(WORK),a0
355:            tst.b   (a0)
356:            bne     setfn0
357:            lea.l   dicext(pc),a0
358: setfn0:    STRCPY
359:            movem.l (sp)+,a0-a1
360:            rts
361: *
362: nextarg:
363:            bsr     skipsp
364: .ifdef DOSLIKE
365:            cmpi.b  #'/',(a2)
366:            beq     nxarg0
367: .endif
368:            cmpi.b  #'-',(a2)
369:            beq     nxarg0
370:            rts
371: nxarg0:    addq.l  #1,a2
372:            move.b  (a2)+,d0
373:            bclr.l  #5,d0
374:            cmpi.b  #'R',d0
375:            bne     usage
376:            moveq.l #-1,RFLAGREG
377:            bra     nextarg
378:
379: *
380: getarg:
381:            move.l  a0,-(sp)
382: gtarg0:    tst.b   (a2)
383:            beq     gtarg1
384:            cmpi.b  #SPACE,(a2)
385:            beq     gtarg1
386:            cmpi.b  #TAB,(a2)
387:            beq     gtarg1
388:            cmpi.b  #'-',(a2)
389:            beq     gtarg1
390: .ifdef DOSLIKE
391:            cmpi.b  #'/',(a2)
392:            beq     gtarg1
393: .endif
394:            move.b  (a2)+,(a0)+
395:            bra     gtarg0
396: gtarg1:    clr.b   (a0)
397:            movea.l (sp)+,a0
398:            rts
399: *
400: skpsp0:    addq.l  #1,a2
401: skipsp:    cmpi.b  #SPACE,(a2)
402:            beq     skpsp0
403:            cmpi.b  #TAB,(a2)
404:            beq     skpsp0
405:            rts
406: *
407:            .data
408: *
409: ttlmes:    .dc.b   'ASK68K Dictionary File Selector v1.00'
410:            .dc.b   CR,LF,0
411: mes1:      .dc.b   'CHDICを切り放しました',CR,LF,0
412: errms0:    .dc.b   'メモリ不足です',CR,LF,0
413: errms1:    .dc.b   'CHDICはすでに組み込み済みです',CR,LF,0
414: errms2:    .dc.b   'CHDICはまだ組み込まれていません',CR,LF,0
415: usgmes:    .dc.b   '使用法:CHDIC [スイッチ] 辞書ファイル […]'
416:            .dc.b   CR,LF
417:            .dc.b   TAB,'/R',TAB,'CHDICの常駐を解除する',CR,LF,0
418: dicext:    .dc.b   '.DIC',0
419: *
420:            .end
```

▶もう過ぎ去ってしまったX68000のウイルスですが，先日本屋で横にいた中学生の話が耳に入ってきたのですが，中学生A「日本でいちばんウイルスが入りやすいパソコンてX68000でしょうー」，中学生B「そーそー，オレX68000，大丈夫かなー？」と，とんでもない印象を受けた人もいるようです。　　前多　和洋（16）愛知県

X68000の画像を読み込むには

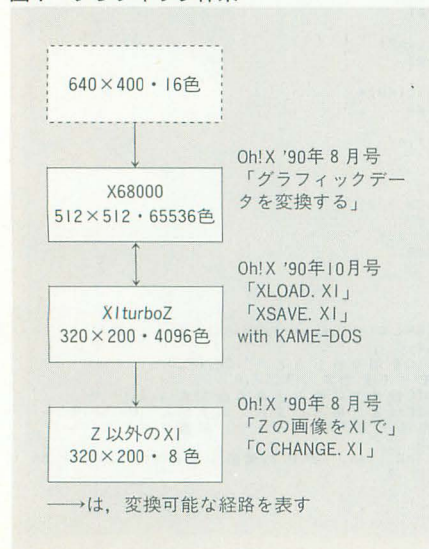
亀田 雅彦 Kameda Masahiko

前回はX1turboZの画像をX1に持ってくる話をしました。今回は一歩進んでX68000の65536色の画像データをX1turboで表示してみましょう。MS-DOSフォーマットのディスクが読めるということはテキストデータの互換だけではないのです。

8月号の特集では4096色と8色がらみのグラフィック変換に取り組みました。今月はなんとX68000とX1turboZ（またはX1turboで2HDドライブを装備したもの）でグラフィックデータの相互変換をするためのプログラムです。図1のようなグラフィック体系が完成されますね。こんなインタマシン的なものが好きなんですよ。

それから今月は「2HDドライブを持っているX1turboユーザー」にしか使えません。汎用的な外部コマンドの作成は来月以降となりますのでご了承を。ただし，X68000とのデータ互換なのでX68000ユーザーには非常に関係のあるところです。これまであまり関係のない記事と思っていた方もちょっと目を止めてみてください。

データの共有化

あなたはどこでグラフィックデータを手に入れていますか？ パソコン通信，スキャナ，自分でツールを使って描くなど，いろんな方法があります。また，こうしたデータを使ってゲームなどを作ることもあるでしょう。そんなとき，簡単に画像を手に入れることができるものとしてビデオデジタイザがあります。これはテレビやビデオの画像を一瞬のうちにG-RAMに取り込んでしまうものです。X68000にもカラーイメージユニットなる機械がありますね。X1turboZにはなんと標準でこのデジタイズ機能が内蔵されているのです。

X68000での画像データの美しさは定評がありますし，X1turboZのビデオデジタイザの機能と性能は素晴らしいものがあります。なんとX68000のカラーイメージユニットを凌ぐところもあります。

そこでXLOAD.X1，XSAVE.X1の登場です。8月号のGLOAD.X1，GSAVE.X1に似ていますが，実は機能も似ています。対象になるグラフィックデータファイルがX68000のGL3ファイル形式という違いだけです。機能は，それぞれ，「X68000でセーブした65536色のデータをX1turboZの4096色や8色疑似中間調で表示する」「X1turboZの4096色画像をX68000で読めるかたちに変換する」というものです。

図1　グラフィック体系

入力方法

今回はturboBASICあるいはZ-BASIC専用のプログラムなのでKAME-DOSもそれらのBASIC上で動くものが必要です。7月号と同じように変更をするのですが，もう一度掲載しておくのでよく読んでください。今月のプログラムはリスト1を上記のいずれかのBASICで入力，

SAVE "XLOAD.X1"

でセーブして，

CLEAR &HB000

を実行したあと，リスト2をマシン語入力ツールなどで入力して，

SAVEM"X681.OBJ",&HB000,&HB6E5

として同一ディレクトリにセーブしてください。

同様に，リスト3を入力し，

SAVE"XSAVE.X1"

でセーブ，リスト4を入力，

SAVEM"X168.OBJ",&HC800,&HCA19

でセーブしてください。

使い方

いまディスク上には，

XLOAD.X1
X681.OBJ
XSAVE.X1
X168.OBJ

の4つのファイルがあると思います。それにX68000のGL3形式の画像データが入っているディスクを用意してください（X68000のグラフィックデータファイルがわからない方は注を参照）。

●X68000ファイル→X1画像

書式：XLOAD ファイル名［オプション］

オプション：

-H	640×400	8色
-M	640×200	8色
-L	320×200	4096色（Z-BASIC）
-L8	320×200	8色

オプション省略時は，-Hを指定したことと同じ。つまり，ひとつのファイルを上記の4つのモードでロードできるということです。XLOAD起動時には，自動的にオプション指定した解像度に設定されます。

そして，書式のようにコマンドラインか

ら起動すると，簡単なメニューが開かれます。

1：ALL screen load
2：PARTS screen load
3：screen CLS
4：QUIT

それぞれの数字を押して選択してください。1は表示画面いっぱいにロードします。3はグラフィック画面クリアで，4は中止です。そして2を押すと，原画を縮小して画面の一部にロードします。

その前に「ロード位置」と「大きさ」を指定してやることになります。白い点が点滅しているので，テンキーで動かしてスペースキーで決定してください。始点と終点を決めると，その矩形領域に画像をロードします。左上の数字は白い点の座標です。

これを使えば複数のグラフィックを合成できますし，ゲームなどで使いやすい大きさに変換することもできます。

具体的に例を挙げておきましょう。まず，KAME-DOSを起動します。Bドライブに画像ディスクを入れてコマンドラインから，

XLOAD B:ファイル名.GL3　-H

とすると高解像度で立ち上がります。メニューから1を選んでください。全画面ロードが始まります。画面にゆっくり画像を表示していき，数分で完了してスペースキー入力待ちになります。

●X1画像→X68000ファイル

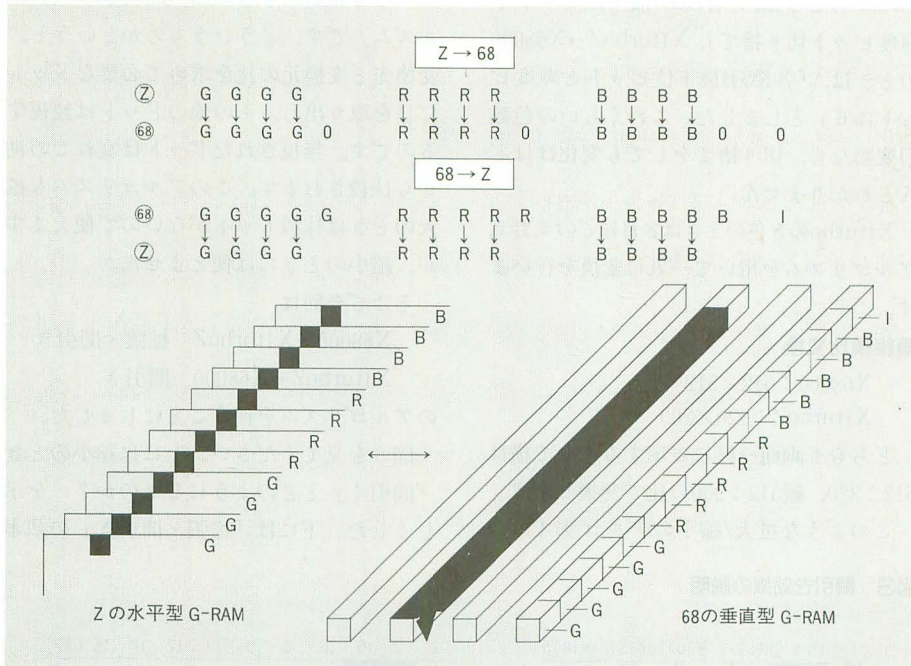
書式：XSAVE ファイル名

セーブはZ-BASIC専用でして，

320×200　4096色→512×512 65536色

という変換を行います。具体例としては，まず，グラフィックを表示しておいて，

XSAVE B:ファイル名.GL3

でXSAVEを起動すると，そのままグラフィックをX68000の形式でセーブします。左上の数字は終了までのライン数です。

なお，セーブ・ロードどちらの場合もディスクがMS-DOSフォーマットでファイル拡張子がGL3でないと動作しないので注意してください。

●画像の違い

XLOADするときには，前述した4つのモードが選べます。そのなかで，もし解像度の決まっていない変換をするのなら「640×400　8色モード」をおすすめします。X68000は512×512という高解像度なので，その他のモードだと切り捨てられる情報が多すぎるのです。特にアニメ調の絵の場合，使われている色数が少ないためか，輪郭の太さでかなり印象が違います。320×200ではMSX調の絵になるのが許せません。

逆に色の変化を大切にしたり輪郭線がないような絵なら，4096色を使うことになるでしょう。用途にあわせて比較検討してみてください。

図2　色変換の説明

ビット切り捨て&カサ上げ

X68000とX1turboでグラフィックを相互変換しようとすると画面の縦横比と色数という2段変換が必要となります。今回はX68000の512×512ドット65536色とX1turboZの320×200ドット4096色という固定画像を基本として，X1turboの640×400，640×200，320×200ドット8色モードまでサポートしています。

●色の変換

X68000のG-RAMはご存じのように垂直型です。画面上の1ドットは1ワードで，

GGGGG RRRRR BBBBB I

というふうにRGBそれぞれに5ビット，輝度に1ビット（5×3＋1＝16ビット）となるわけです。

X1turboZのG-RAMは水平型なので画面上の1ドットが1ビットに当たります。それが12枚重なって，$2^{12}=4096$色です。便宜的に垂直型のように書くと，

GGGG RRRR BBBB

となります。

2つのG-RAM構造を比べてみると「各RGBについて1ビットずつ違う」ということがわかるでしょう。そこで，X68000→X1

注）X68000のグラフィックデータファイルの作成

●GL3形式のデータファイルの作り方

PIC形式のグラフィックデータをGL3形式にしてみましょう。まずBASICを起動するのですが，そのときに本誌6月号のPIC.FNCを組み込みます。そしてPICファイルをBドライブに入れて，

```
SCREEN 1,3,1,1
PIC_LOAD（"B：ファイル名"）
```

とします。これで画面にロードできました。次に画面をセーブします。Bドライブに空のディスクをセットしてください。グラフィックを表示させたまま，

```
IMG_SAVE（"B：ファイル名"）
```

とします（このとき拡張子はいりません）。これでBドライブにGL3形式のファイルが出来上がりました。

●GL3形式のファイルをロードする

BASICを起動して，Bドライブにファイルの入ったディスクを入れて，

```
SCREEN 1,3,1,1
IMG_LOAD（"B：ファイル名"）
```

で，ロードできます。

turboZのときは，「各RGB最下位ビットと輝度ビット切り捨て」，X1turboZ→X68000のときは，「各RGB最下位ビットと輝度ビットは0」としました。これくらいの色数の変換なら，切り捨てをしても変化はほとんどわかりません。

X1turboの8色のときは8月号での桒野式アルゴリズムを用いて一気に変換を行います。

●縦横比変換

X68000：512×512
X1turboZ：320×200

どちらも画面一杯に表示するとして横は512：320，縦512：200の比で変換します。

このような拡大/縮小のアルゴリズムのもっとも原始的なものは「間引きのアルゴリズム」です。どういうものかというと，変換先と変換元の比を求めて必要なドットだけを取り出し，その他のドットは無視するのです。無視されたドットは哀れこの世から抹殺されます。このアルゴリズムも拡大のときは抹殺ドットがないので使えますが，縮小のときには使えません。

そこで今回は，

X68000→X1turboZ　拡張・間引き
X1turboZ→X68000　間引き

のアルゴリズムを使うことにしました。

図3を見てください。上には縮小のとき「間引く」とどのようになるのか？　を示しました。下には「拡張・間引き」の基本的な考え方があります。つまり「ふつうの間引きでは無視されてしまうドット情報を表示するドットに持たせてやる」というものです。こうすればラインがプチプチ切れてしまうことがありません（ちなみにZ'sSTAFF-Zでやったときはプチプチ切れてしまった）。

そして，そこで問題になってくるのが「どうやって情報を持たせるか？」です。図の最後のほうに書いてあるように，すべての色情報を「OR」するのが簡単ですが，これだとカラーの場合に色が変化します。

たとえば，赤と青の点が並んでいると表示色はマゼンタになってしまいます。それでいい場合もあるのですが，今回は採用しませんでした。その代わり，「表示ドットを固定しない」方針でのぞみました。

図3　間引き効果の説明

図4を見てください。表示色をどのように決定するかを示しました。まず色を背景色（ここではカラーコード0）と表示色（背景色以外）に分けます。そして，「優先色は普通色より優先」「遠いドットより近いドット優先」という優先順位をつけます。これに従って，実際の表示色が決定されます。遠いドットから順にサーチされて，そのなかのひとつのドットの色が表示色となるのです。

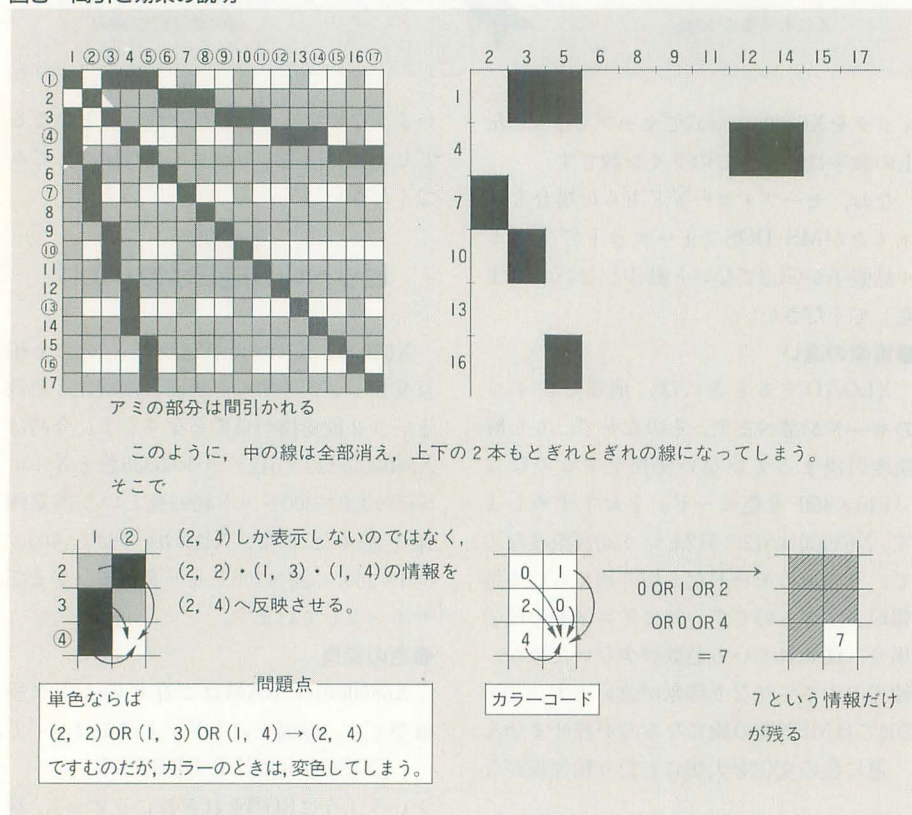
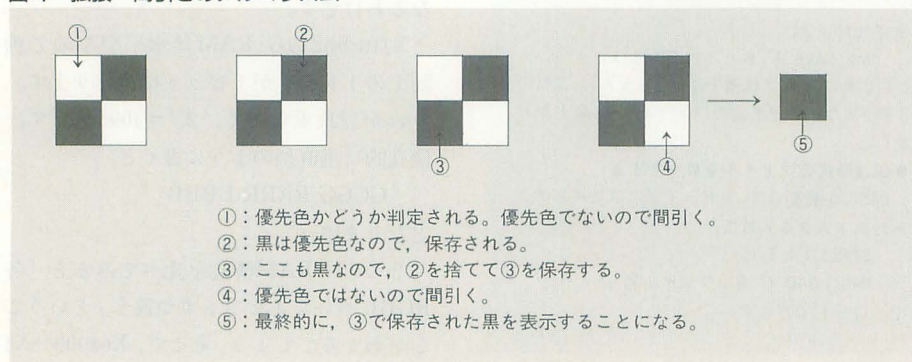
このアルゴリズムの意味は，「オーバーサンプリングされたドット内でもっとも目立ったドットを表示する」ことです。絵の印象は「絵の中のキーポイントで決定する」という仮定の下に構築されました。デジタイズ画面が変換対象なので実はあまり関係ないのですが，こういう方法もあるということです。

図4　拡張・間引きのアルゴリズム

優先色

このプログラムをこのまま使うと優先色は黒（カラーコード0）だけになってしまいます。ほかの色を優先色にしたい場合には&HB31E, &HB6E0からの0をワード単位で書き換えてください。この数字が背景色のカラーコード（X68000のもの）になります。同時に3個まで指定できます。

たとえば，背景を白（カラーコード&HFFFF）で描いて輪郭を粗いラインで描いたグラフィックを転送するときは，

FOR I=0 TO 5 STEP 2

POKE &HB31E+I,&HFF,&HFF

POKE &HB6E0+I,&HFF,&HFF

NEXT

のようにして優先色を白１色にしておきましょう。

なお，これはX68000→X1turboZの場合だけでX1turboZ→X68000のときは必要ありません。

＊　＊　＊

拡大/縮小アルゴリズムにはいろいろなバリエーションがあるのでどれがいいとは一概にいえません。しかも，Z80に過重な負担をかけるようなアルゴリズムにもできないので困りました。

ただ，X1turboZのグラフィックまわりには量子化，モザイク，クロマキーなど，X68000のオプションですら備えていないような機能が標準装備されています。それを見捨ててしまうのはあまりにももったいないと思いませんか？　ちょっと語弊があるかもしれませんが，X1turboZなら「安価なインテリジェントデジタイザ」として使ってもよいのではないかと思います。

＊　＊　＊

Ｘ１とX68000。同じ「X」の名を冠しながら，２機種間の交流には通信回線などを必要としていました。X1時代に蓄えられた資産（データやプログラムのこと）は決して少なくありません。これらが古いフロッピーディスクとともに闇に葬られていたのでは資源の無駄です。そこでこのKAME-DOSの発想が生まれました。「過去の資源を最大限に利用して現在の効用を最大にすること」これが第１です。

そして第２に，「８ビットは８ビットなりの，16ビットは16ビットなりの役割分担をさせる」ことです。それぞれの機種の特性（値段も含めた）を生かした活動をし，互いに情報交換できれば２倍の効用が得られるでしょう。たとえば，「MusicBASICで作った音色をX68000で使いたい」といったときもファイルコピー一発でできます。「カラーイメージユニットがないけどイメージ取り込みがしたい」場合，X1turboZがあればそれを生かすことができるわけです。

Ｘ１のコマンドシェルシミュレータとしてだけでなくFDを介したデータコンバート機能も存分に活用してください。

リスト1　XLOAD.X1

```
1000 'XLOAD.X1  Ver 1.1        By Kameda
1010 '
1020 DEFINT a-z:KLIST 0:CONSOLE 0,25
1030 DEFUSR1=m_opens:DEFUSR2=m_preop
1040 RESTORE 1050:FOR i=1 TO 9:READ x(i),y(i):NEXT
1050 DATA -8,8,0,8,8,8,-8,0,0,0,8,0,-8,-8,0,-8,8,-8
1060 '
1070 GOSUB "select":GOSUB "box":INIT:IF k=4 GOTO 1270
1080 IF mc=0 THEN PRW 255 ELSE GOSUB 1950
1090 POKE v_wfd0,PEEK(&HF8D6):CLEAR &HB000:LOADM "X681.OBJ"
1100 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1110 MEM$(s_ff,2)=MKI$(&H2000):MEM$(v_badr,2)=MKI$(&H3000)
1120 bsiz!=&H10*&H100:MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,1
1130 POKE v_dn,PEEK(s_dn):IF fe$(1)="" THEN "!2"
1140 '----------( MAIN ROUTINE )------------
1150 '
1160 GOSUB 1440
1170 K=PEEK(v_stop):IF k=3 THEN "!3" ELSE IF k<>0 THEN "!"
1180 IF PEEK(v_mac)<>2 THEN "!4"
1190 f$=MEM$(v_fnam+13,3):IF f$<>"GL3" AND f$<>"gl3" THEN "!4"
1200 MEM$(x681+&H1D,8)=MKI$(sx)+MKI$(sy)+MKI$(ex)+MKI$(ey):POKE &HB025,HML
1210 '
1220 CALL m_devi:IF PEEK(v_stop) THEN "!"
1230 CALL x681
1240 IF PEEK(v_iofg) GOTO 1210
1250 '
1260 CLS:INIT:KLIST 0:GOSUB "ending"
1270 POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$:CLEAR &HD000
1280 KLIST 1:proces=proces-1:CHAIN proces$(proces)
1290 '----------(    SELECT     )-----------
1300 '
1310 LABEL "select"
1320 mc=1:IF op$="-L" OR op$="-l" GOTO 1410
1330 mc=0:IF op$="-M" OR op$="-m" GOTO 1370
1340 IF op$="-L8" OR op$="-l8" GOTO 1390
1350 x681=&HB000:xr=640:yd=400:c0=7:WIDTH 80,25,1,2:OPTIONSCREEN 4:INIT
1360 HML=0:RETURN
1370 x681=&HB000:xr=640:yd=200:c0=7:WIDTH 80,25,0,1:OPTIONSCREEN 4:INIT
1380 HML=1:RETURN
1390 x681=&HB000:xr=320:yd=200:c0=7:WIDTH 40,25,0,1:OPTIONSCREEN 4:INIT
1400 HML=1:RETURN
1410 x681=&HB400:xr=320:yd=200:c0=4095:OPTIONSCREEN 4:WIDTH 40,25,0,1
1420 OPTIONSCREEN 5:INIT:HML=0:RETURN
1430 '----------(     OPEN      )-----------
1440 '
1450 POKE v_ddrv+1,7,1:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1460 POKE v_od,1:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1470 IF PEEK(v_stop) RETURN
1480 POKE v_sbdr,1:POKE v_op,0:d$=USR1(fe$):RETURN
1490 '----------(     END       )----------
1500 '
1510 LABEL "ending"
1520 CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1530 REPEAT:A$=INKEY$:UNTIL A$=" ":CLS
1540 RETURN
1550 '----------( ERROR ROUTINE )----------
1560 '
1570 LABEL "!4":RESTORE "m3":GOTO 1610
1580 LABEL "!3":RESTORE "m2":GOTO 1610
1590 LABEL "!2":RESTORE "m1":GOTO 1610
1600 LABEL "!" :RESTORE "m0"
```

```
1610 READ m$:BEEP:CLS:INIT:CREV 1:PRINT m$;:CREV 0:PRINT
1620 POKE v_stop,0:GOTO 1270
1630 '----------(       BOX       )----------
1640 '
1650 LABEL "box"
1660 s=0
1670 COLOR 3:LOCATE 3,8:PRINT "push [1]-[4] or SPACE":COLOR 7
1680 LOCATE 5,10:PRINT " 1:ALL screen load"
1690 LOCATE 5,12:PRINT " 2:PARTS screen load"
1700 LOCATE 5,14:PRINT " 3:screen CLS"
1710 LOCATE 5,16:PRINT " 4:QUIT"
1720 REPEAT:a$=INKEY$(1):k=VAL(a$):UNTIL (1<=k AND k<=4) OR a$=SPACE$(1):CLS
1730 IF k=1 THEN sx=0:sy=0:ex=xr:ey=yd:RETURN
1740 IF k=2 GOTO 1780
1750 IF k=3 THEN CLS 0:GOTO 1670
1760 IF k=4 RETURN
1770 IF a$=SPACE$(1) AND s=0 THEN SCREEN:s=1:GOTO 1670 ELSE INIT:s=0:GOTO 1670
1780 '
1790 INIT:x=0:y=0:GOSUB 1860:IF a$=CHR$(27) GOTO "box"
1800 sx=x:sy=y:PSET (x,y,c0)
1810 x=xr/2:y=yd/2:GOSUB 1860:ex=x:ey=y
1820 x=sx:y=sy:LINE (x,y)-(x,y),XOR,c0:IF a$=CHR$(27) GOTO "box"
1830 IF ex<sx THEN SWAP ex,sx
1840 IF ey<sy THEN SWAP ey,sy
1850 CLS:LINE (sx,sy)-(ex-1,ey-1),PSET,c0,b:RETURN
1860 '
1870 LINE (x,y)-(x,y),XOR,c0:LOCATE 0,0:PRINT "(x,y)=";
1880 '
1890 LOCATE 6,0:PRINTUSING "####",x,y;:KEY0,""
1900 a$=INKEY$(0):a=VAL(a$):LINE (x,y)-(x,y),XOR,c0
1910 w=x+x(a):IF 0<=w AND w<=xr THEN x=w
1920 w=y+y(a):IF 0<=w AND w<=yd THEN y=w
1930 IF a$=SPACE$(1) OR a$=CHR$(27) THEN KEY0,"":RETURN
1940 LINE (x,y)-(x,y),XOR,c0:GOTO 1890
1950 '
1960 SCREEN 0,0,1:FOR i=1 TO 7:GOTO i,0:NEXT:RETURN
1970 '----------( DATA AREA       )----------
1980 LABEL "m0":DATA エラーが発生しました！！
1990 'LABEL "m0":DATA Error !!
2000 LABEL "m1":DATA ファイル・ネームを指定してください
2010 'LABEL "m1":DATA Need FILE-NAME
2020 LABEL "m2":DATA ファイル・ネームが違います
2030 'LABEL "m2":DATA FILE Not Found
2040 LABEL "m3":DATA ファイルまたはディスクが違います
2050 'LABEL "m3":DATA FILE or DISK Error
```

リスト2 XSAVE.X1

```
1000 'XSAVE.X1  Ver 1.0        By Kameda
1010 '
1020 OPTIONSCREEN 5:INIT:DEFINT a-z:KLIST 0
1030 DEFUSR1=m_opens:DEFUSR2=m_preop
1040 '
1050 POKE v_wfd0,PEEK(&HF8D6):CLEAR &HC000:LOADM "X168.OBJ"
1060 iomm=PEEK(v_iomm):badr$=MEM$(v_badr,2):ff$=MEM$(v_ff,2)
1070 MEM$(s_ff,2)=MKI$(&H6000):MEM$(v_badr,2)=MKI$(&H7000)
1080 bsiz!=&H10*&H100:MEM$(v_bsiz,2)=MKI$(bsiz!):POKE v_iomm,4
1090 POKE v_dn,PEEK(s_dn):IF fe$(1)="" THEN "!2"
1100 '----------( MAIN ROUTINE )------------
1110 '
1120 GOSUB 1340:x168=&HC800:IF PEEK(v_stop)<>0 THEN "!"
1130 IF PEEK(v_mac)<>2 THEN "!3"
1140 f$=MEM$(v_fnam+13,3):IF f$<>"GL3" AND f$<>"gl3" THEN "!3"
1150 '
1160 i=1:CLS
1170 REPEAT
1180   CALL x168:LOCATE 0,0:PRINT i;"/128"
1190   POKE v_od,2:POKE v_iofg,2:POKE v_edr,0:CALL m_devi
1200   i=i+1
1210 UNTIL i=128 OR PEEK(v_stop)
1220 IF PEEK(v_stop) THEN "!"
1230 CALL x168
1240 POKE v_od,2:POKE v_iofg,2:POKE v_edr,2:CALL m_devi
1250 IF PEEK(v_stop) THEN "!"
1260 '
1270 POKE v_zoku+1,0,0:MEM$(v_fszl,4)=MKI$(&H0)+MKI$(&H8)
1280 POKE v_od,2:MEM$(v_badr,2)=MKI$(&H3000):CALL m_saved:CLS
1290 '
1300 GOSUB "ending":CLEAR &HD000
1310 POKE v_iomm,iomm:MEM$(s_ff,2)=ff$:MEM$(v_badr,2)=badr$
1320 proces=proces-1:CHAIN proces$(proces)
1330 '----------(    OPEN     )-----------
1340 '
1350 POKE v_ddrv+1,1,7:POKE v_iofg,0:POKE s_escp,0:fe$=fe$(1)
1360 POKE v_od,2:d$=USR2(fe$):fe$=RIGHT$(fe$,PEEK(v_yen))
1370 IF PEEK(v_stop) RETURN
1380 POKE v_sbdr,1:POKE v_op,3:d$=USR1(fe$):RETURN
1390 '----------(     END      )-----------
1400 '
1410 LABEL "ending"
1420 CFLASH 1:PRINT "PUSH SPACE":CFLASH 0
1430 REPEAT:a$=INKEY$:UNTIL a$=" "
1440 RETURN
1450 '----------( ERROR ROUTINE )----------
1460 '
1470 LABEL "!3":RESTORE "m2":GOTO 1500
1480 LABEL "!2":RESTORE "m1":GOTO 1500
1490 LABEL "!" :RESTORE "m0":OUT &HFFC,PEEK(v_dn)
1500 READ m$:BEEP:CLS:CREV 1:PRINT m$;:CREV 0:PRINT
1510 POKE v_stop,0:GOTO 1290
1520 '----------( DATA AREA      )----------
1530 LABEL "m0":DATA エラーが発生しました！！
1540 'LABEL "m1":DATA Error !!
1550 LABEL "m1":DATA ファイル・ネームを指定してください
1560 'LABEL "m1":DATA Need FILE-NAME
1570 LABEL "m2":DATA ディスクまたはファイルが違います
1580 'LABEL "m2":DATA DISK or FILE-NAME Error
```

リスト3 X681.OBJ

```
B000 C3 36 B0 00 00 00 00 00 : A9
B008 00 00 00 00 00 00 00 00 : 00
B010 00 00 00 00 00 00 00 00 : 00
B018 00 00 02 00 02 00 00 00 : 04
B020 00 80 02 90 01 00 00 00 : 13
B028 00 00 00 00 00 00 00 00 : 00
B030 00 00 00 00 00 00 E1 22 : 03
B038 2F B0 3A 35 B0 B7 C2 38 : AF
B040 B1 3E 01 32 35 B0 21 00 : 28
B048 B8 11 01 B8 01 FF 0F AF : 40
B050 77 ED B0 21 00 00 22 2A : 81
B058 B0 2A 19 B0 ED 5B 15 B0 : B0
B060 B7 ED 52 22 0B B0 2A 1B : 18
B068 B0 ED 5B 17 B0 B7 ED 52 : B5
B070 22 0D B0 2A 21 B0 ED 5B : 22
B078 1D B0 B7 ED 52 22 0F B0 : A4
---------------------------------
SUM: 28 63 CD D0 04 FA 1D 5B 426C

B080 B7 CB 1C CB 1D CB 1C CB : 38
B088 1D CB 1C CB 1D 7D 32 2C : C7
B090 B0 2A 23 B0 ED 5B 1F B0 : C4
B098 B7 ED 52 22 11 B0 2A 17 : 1A
B0A0 B0 22 05 B0 2A 0D B0 29 : 97
B0A8 ED 5B 11 B0 B7 ED 52 22 : 21
B0B0 13 B0 2A 1F B0 22 09 B0 : 97
B0B8 AF 32 34 B0 3A 14 B0 FE : C1
B0C0 80 D2 56 B1 2A 15 B0 22 : 6A
B0C8 03 B0 2A 0B B0 29 ED 5B : 09
B0D0 0F B0 B7 ED 52 44 4D 2A : 70
B0D8 1D B0 22 07 B0 CD 05 B3 : 2B
B0E0 AF 32 33 B0 2A 07 B0 29 : CE
B0E8 11 00 CB 19 22 31 B0 3E : 36
B0F0 7F B8 38 12 CD 9D B1 2A : C6
B0F8 0F B0 29 50 59 EB B7 ED : 20
---------------------------------
SUM: 97 88 D9 72 51 92 09 8F 1D5E

B100 52 44 4D C3 EF B0 2A 0B : 7A
B108 B0 29 09 44 4D 3A 33 B0 : 90
B110 21 34 B0 B6 CC 8C B1 2A : EE
B118 07 B0 23 22 07 B0 ED 5B : FB
B120 21 B0 B7 ED 52 DA E0 B0 : 31
B128 2A 2A B0 23 22 2A B0 11 : 34
B130 04 00 B7 ED 52 CA 81 B1 : F6
```

▶僕の友達に「65536色とスプライトとFM音源とジョイスティックが標準装備のパソコンが，学校の勉強でどーしても必要なんだ」といって，何も知らない親にPROを買ってもらったとんでもないやつがいます。ちなみに学校ではPascalをやっています(笑)。皆さんも参考にしてください。
吉永　孝明（19）福岡県

```
B138 2A 05 B0 23 22 05 B0 2A : 03
B140 11 B0 29 ED 5B 13 B0 EB : E0
B148 B7 ED 52 22 13 B0 3E 01 : 1A
B150 32 34 B0 C3 BC B0 2A 0D : 7C
B158 B0 29 ED 5B 13 B0 19 22 : 1F
B160 13 B0 CD FB B1 2A 09 B0 : 1F
B168 23 22 09 B0 ED 5B 23 B0 : 19
B170 B7 ED 52 DA B8 B0 21 00 : 59
B178 00 22 13 B0 2A 2F B0 E5 : D3
---------------------------------
SUM: 3A 0B 4A 61 B4 80 EA 3C 8A57

B180 C9 21 00 00 22 2A B0 2A : 10
B188 2F B0 E5 C9 DD 2A 31 B0 : 75
B190 DD 7E FE DD 77 00 DD 7E : 08
B198 FF DD 77 01 C9 C5 DD 2A : E9
B1A0 31 B0 2A 28 B0 CD 12 E0 : A2
B1A8 47 23 CD 12 E0 4F 23 22 : BD
B1B0 28 B0 3A 34 B0 B7 C2 C7 : 36
B1B8 B1 3A 33 B0 B7 C2 C7 B1 : BF
B1C0 3C 32 33 B0 C3 F3 B1 DD : 95
B1C8 66 00 DD 6E 01 B7 ED 42 : 98
B1D0 28 27 09 EB FD 21 1E B3 : 32
B1D8 3E 03 FD 6E 00 FD 66 01 : 10
B1E0 B7 ED 52 28 14 19 B7 ED : EF
B1E8 42 28 08 FD 23 FD 23 3D : EF
B1F0 C2 DA B1 DD 70 00 DD 71 : E8
B1F8 01 C1 C9 2A 1D B0 29 01 : AC
---------------------------------
SUM: E9 F5 A8 68 BB 3C 5B 6B B9BB

B200 00 CB 09 22 2D B0 CD EB : 8B
B208 B2 FD 21 01 B8 DD 21 01 : 88
B210 C0 ED 5B 1D B0 DD 19 FD : C8
B218 19 01 D0 1F ED 78 E6 EF : 43
B220 ED 79 3A 09 B0 E6 01 28 : 68
B228 14 DD E5 FD E5 DD E1 FD : 73
B230 E1 3A 25 B0 B7 20 06 ED : BA
B238 78 F6 10 ED 79 3E C0 CD : AF
B240 4C B2 3E 80 CD 4C B2 3E : C5
B248 40 C3 4C B2 ED 4B 26 B0 : 0F
B250 80 47 2A 2D B0 3A 2C B0 : E4
B258 5F DD E5 FD E5 D5 16 08 : F6
B260 1E 00 CD 86 B2 CD AF B2 : 51
B268 CB 13 DD 23 FD 23 15 C2 : D5
B270 62 B2 ED 59 03 D1 1D C2 : 0D
B278 5D B2 FD E1 DD E1 11 82 : 3E
---------------------------------
SUM: F8 4C D6 41 25 4B A1 15 5F47

B280 02 FD 19 DD 19 C9 78 FE : 4D
B288 80 38 1C FE C0 38 06 7E : 4E
B290 23 23 E6 F8 C9 C5 46 23 : 1B
B298 7E 23 CB 18 1F CB 18 1F : A5
B2A0 CB 18 1F E6 F8 C1 C9 23 : 8D
B2A8 7E 23 E6 3E 07 07 C9 FD : 99
B2B0 86 00 38 05 FE F8 3F 30 : 28
B2B8 03 D6 F8 37 F5 CD C2 B2 : 3E
B2C0 F1 C9 E5 67 CB 3F CB 3F : 1A
B2C8 CB 3F 6F DD 77 01 7C E6 : 30
B2D0 07 DD 86 00 85 DD 77 00 : 43
B2D8 7D 87 DD 86 FF DD 77 FF : B9
B2E0 7D 87 87 FD 86 01 FD 77 : 83
B2E8 01 E1 C9 2A 1D B0 ED 5B : EA
B2F0 09 B0 AF 32 F6 FB 06 1D : AE
B2F8 ED 41 CD 07 59 06 1E ED : 6C
---------------------------------
SUM: A9 51 9E 75 6B CA B2 C0 C53F

B300 41 22 26 B0 C9 2A 2A B0 : 06
B308 11 00 04 CD 27 E0 ED 5B : 31
B310 03 B0 EB 29 19 ED 5B A4 : CC
B318 E0 19 22 28 B0 C9 00 00 : BC
B320 00 00 00 00 08 66 0D DC : 57
B328 11 50 16 C3 1A 33 1F A1 : 47
B330 23 0C 28 75 2C D9 30 3A : 3B
B338 35 97 39 EF 3D 42 42 91 : 46
B340 46 D9 4A 1C 4F 59 53 8F : 0F
B348 57 BE 5B E7 5F 07 64 20 : 41
B350 68 2F 6C 39 70 39 74 30 : 89
B358 78 1D 7C 00 80 DA 83 A9 : 97
B360 87 6E 8B 28 8F D6 92 79 : 18
B368 96 11 9A 9C 9D 1B A1 8E : C4
B370 A4 F4 A7 4C AB 98 AE F0 : 6C
B378 B1 8B B5 40 B8 50 BB 3F : 33
---------------------------------
SUM: 8D BF BC 81 71 C0 5A B5 E049

B380 BE 35 C1 1C C4 F3 C6 BB : 08
B388 C9 74 CC 1C CF B4 D1 3C : B5
B390 D4 B3 D6 1A D9 70 DB B4 : 4F
B398 DD 40 DF E0 E1 97 E4 1B : 53
B3A0 E6 04 E8 DE E9 A7 EB 5C : 87
B3A8 ED FF EE 90 F0 0E F2 79 : D3
B3B0 F3 D1 F4 16 F6 47 F7 66 : 68
B3B8 F8 71 F9 68 FA 4C FB 1D : 28
B3C0 FC D9 FC 83 FD 18 FE 99 : 00
B3C8 FE 07 FF 61 FF A7 FF D8 : E2
B3D0 FF F6 FF 00 00 3E 01 32 : 65
B3D8 53 FE 21 FF FF 22 54 FE : E4
B3E0 2A 5D FE 22 03 FF 22 05 : D0
B3E8 FF 21 00 00 22 07 FF ED : 35
B3F0 5B 63 FE AF CB 7A 28 07 : DF
B3F8 3C 21 00 00 ED 52 EB 32 : B9
---------------------------------
SUM: 02 B7 1C D2 EE E7 AB EA 632D

B400 C3 35 B4 00 00 00 00 00 : AC
B408 00 00 00 00 00 00 00 00 : 00
B410 00 00 00 00 00 00 00 00 : 00
B418 00 00 02 00 02 00 00 00 : 04
B420 00 40 01 C8 00 00 00 00 : 09
B428 00 00 00 00 00 00 00 00 : 00
B430 00 00 00 00 00 E1 22 2E : 31
B438 B4 2A 13 B4 7C B5 C2 1C : B4
B440 B5 21 00 00 22 29 B4 2A : FF
B448 19 B4 ED 5B 15 B4 B7 ED : 82
B450 52 22 0B B4 2A 1B B4 ED : 19
B458 5B 17 B4 B7 ED 52 22 0D : 4B
B460 B4 2A 21 B4 ED 5B 1D B4 : CC
B468 B7 ED 52 22 0F B4 B7 CB : 5D
B470 1C CB 1D CB 1C CB 1D CB : 9E
B478 1C CB 1D 7D 32 2B B4 2A : BC
---------------------------------
SUM: 95 5A 23 60 16 E5 CA CF 699B

B480 23 B4 ED 5B 1F B4 B7 ED : 96
B488 52 22 11 B4 2A 17 B4 22 : 50
B490 05 B4 2A 0D B4 29 ED 5B : 15
B498 11 B4 B7 ED 52 22 13 B4 : A4
B4A0 2A 1F B4 22 09 B4 AF 32 : BD
B4A8 33 B4 3A 14 B4 FE 80 D2 : 39
B4B0 3A B5 2A 15 B4 22 03 B4 : BB
B4B8 2A 0B B4 29 ED 5B 0F B4 : 1D
B4C0 B7 ED 52 44 4D 2A 1D B4 : 82
B4C8 22 07 B4 CD C7 B6 AF 32 : 08
B4D0 32 B4 2A 07 B4 29 11 00 : 05
B4D8 CD 19 22 30 B4 3E 7F B8 : 61
B4E0 38 12 CD 70 B5 2A 0F B4 : 29
B4E8 29 50 59 EB B7 ED 52 44 : F7
B4F0 4D C3 DD B4 2A 0B B4 29 : B3
B4F8 09 44 4D 2A 07 B4 23 22 : C4
---------------------------------
SUM: DB FB 4D FE C6 62 40 6B CB36

B500 07 B4 ED 5B 21 B4 B7 ED : 7C
B508 52 C2 CE B4 2A 29 B4 23 : C0
B510 22 29 B4 11 04 00 B7 ED : B8
B518 52 CA 65 B5 2A 05 B4 23 : 3C
B520 22 05 B4 2A 11 B4 29 ED : E0
B528 5B 13 B4 EB B7 ED 52 22 : 25
B530 13 B4 3E 01 32 33 B4 C3 : E2
B538 AA B4 2A 0D B4 29 ED 5B : BA
B540 13 B4 19 22 13 B4 CD D4 : 6A
B548 B5 2A 09 B4 23 22 09 B4 : 9E
B550 ED 5B 23 B4 B7 ED 52 C2 : D7
B558 A6 B4 21 00 00 22 13 B4 : 64
B560 2A 2E B4 E5 C9 21 00 00 : DB
B568 22 29 B4 2A 2E B4 E5 C9 : B9
B570 C5 DD 2A 30 B4 2A 27 B4 : B5
B578 CD 12 E0 47 23 CD 12 E0 : E8
---------------------------------
SUM: 40 1C 7C 08 E2 90 4B A8 BBF4

B580 4F 23 22 27 B4 3A 33 B4 : 90
B588 B7 C2 A0 B5 3A 32 B4 B7 : A5
B590 C2 A0 B5 DD 70 00 DD 71 : B2
B598 01 3C 32 32 B4 C3 D2 B5 : 9F
B5A0 DD 66 00 DD 6E 01 B7 ED : 33
B5A8 42 28 27 09 EB FD 21 E0 : 83
B5B0 B6 3E 03 FD 6E 00 FD 66 : C5
B5B8 01 B7 ED 52 28 14 19 B7 : 03
B5C0 ED 42 28 08 FD 23 FD 23 : 9F
B5C8 3D C2 B3 B5 DD 70 00 DD : 91
B5D0 71 01 C1 C9 AF 32 34 B4 : C5
B5D8 2A 1D B4 29 01 00 CD 09 : FB
B5E0 22 2C B4 CD AD B6 3E C0 : 30
B5E8 CD FB B5 CD 3B B6 3E 80 : F9
B5F0 CD FB B5 CD 3B B6 3E 40 : B9
B5F8 C3 FB B5 2A 25 B4 84 67 : 61
---------------------------------
SUM: E3 83 E3 60 D3 DC C0 1F 55EE

B600 CD 19 B6 01 D0 1F ED 78 : F1
B608 F6 10 ED 79 CD 19 B6 01 : 09
B610 D0 1F ED 78 E6 EF ED 79 : 8F
B618 C9 E5 CD 6B B6 3A 34 B4 : BE
B620 3C 32 34 B4 FE 04 C2 32 : 4C
B628 B6 ED 5B 2C B4 13 ED 53 : 31
B630 2C B4 7C C6 04 67 CD 6B : C5
B638 B6 E1 C9 DD 2A 2C B4 11 : 58
B640 10 00 3A 2B B4 47 DD CB : 18
B648 00 16 DD CB 02 16 DD CB : 7E
B650 04 16 DD CB 06 16 DD CB : 86
B658 08 16 DD CB 0A 16 DD CB : 8E
B660 0C 16 DD CB 0E 16 DD 19 : E4
B668 10 DC C9 E5 44 4D DD 2A : 32
B670 2C B4 11 10 00 3A 2B B4 : 1A
B678 67 AF DD CB 00 16 17 DD : C8
---------------------------------
SUM: FB 78 96 F7 31 47 64 A7 BBBE

B680 CB 02 16 17 DD CB 04 16 : BC
B688 17 DD CB 06 16 17 DD CB : 9A
B690 08 16 17 DD CB 0A 16 17 : 14
B698 DD CB 0C 16 17 DD CB 0E : 97
B6A0 16 17 ED 79 03 DD 19 25 : B1
B6A8 C2 79 B6 E1 C9 2A 1D B4 : 96
B6B0 ED 5B 09 B4 AF 32 F6 FB : D7
B6B8 06 1D ED 41 CD 07 59 06 : 84
B6C0 1E ED 41 22 25 B4 C9 2A : 3A
B6C8 29 B4 11 00 04 CD 27 E0 : C6
B6D0 ED 5B 03 B4 EB 29 19 ED : 19
B6D8 5B A4 E0 19 22 27 B4 C9 : BE
B6E0 00 00 00 00 00 00       : 00
---------------------------------
SUM: 21 68 D2 4E 53 DA 04 A0 68AA
```

リスト4 X168.OBJ

```
C800 C3 32 C8 00 00 00 00 00 : BD
C808 00 00 00 00 00 00 00 00 : 00
C810 00 00 00 00 00 00 00 00 : 00
C818 00 40 01 C8 00 00 00 00 : 09
C820 00 00 02 00 02 00 00 00 : 04
C828 00 00 00 00 00 00 00 00 : 00
C830 00 00 E1 22 30 C8 2A 13 : 38
C838 C8 7C B5 C2 2B C9 21 00 : D0
C840 00 22 2E C8 2A 19 C8 ED : 10
C848 5B 15 C8 B7 ED 52 22 0B : 5B
C850 C8 2A 1B C8 ED 5B 17 C8 : FC
C858 B7 ED 52 22 0D C8 2A 21 : 38
C860 C8 ED 5B 1D C8 B7 ED 52 : EB
C868 22 0F C8 2A 23 C8 ED 5B : 56
C870 1F C8 B7 ED 52 22 11 C8 : D8
C878 2A 17 C8 22 05 C8 2A 0D : 2F
---------------------------------
SUM: 98 17 66 6B B0 88 8B 76 8B88

C880 C8 29 ED 5B 11 C8 B7 ED : B6
C888 52 22 13 C8 2A 1F C8 22 : 82
C890 09 C8 2A 15 C8 22 03 C8 : C5
C898 2A 0B C8 29 ED 5B 0F C8 : 45
C8A0 B7 ED 52 44 4D 2A 1D C8 : 96
C8A8 22 07 C8 CD F4 C9 CD 69 : B1
C8B0 C9 CD 52 C9 21 00 80 B7 : 09
C8B8 ED 42 38 16 2A 03 C8 23 : 95
C8C0 22 03 C8 2A 0F C8 29 50 : 67
C8C8 59 EB B7 ED 52 44 4D C3 : 8E
C8D0 B4 C8 2A 0B C8 29 09 44 : EF
C8D8 4D 2A 07 C8 23 22 07 C8 : 5A
C8E0 ED 5B 21 C8 B7 ED 52 C2 : E9
C8E8 AE C8 2A 13 C8 11 00 80 : 0C
C8F0 B7 ED 52 D2 0F C9 2A 05 : CF
C8F8 C8 23 22 05 C8 2A 11 C8 : DD
---------------------------------
SUM: 72 34 05 ED 1E A2 D6 D8 9B23

C900 29 ED 5B 13 C8 EB B7 ED : DB
C908 52 22 13 C8 C3 EA C8 2A : EE
C910 0D C8 29 ED 5B 13 C8 19 : 3A
C918 22 13 C8 2A 2E C8 23 22 : 62
C920 2E C8 11 04 00 B7 ED 52 : 01
C928 CA 47 C9 2A 09 C8 23 22 : 1A
C930 09 C8 ED 5B 23 C8 B7 ED : A8
C938 52 C2 92 C8 21 00 00 22 : B1
C940 13 C8 2A 30 C8 E5 C9 21 : CC
C948 00 00 22 2E C8 2A 30 C8 : 3A
C950 E5 C9 ED 5B 29 C8 3A 26 : 47
C958 C8 CD 18 E0 13 3A 25 C8 : C7
C960 CD 18 E0 13 ED 53 29 C8 : 09
C968 C9 C5 CD D6 C9 21 00 00 : 1B
C970 22 25 C8 3E 40 32 2C C8 : B3
C978 CD 9C C9 2A 25 C8 B7 CB : CB
---------------------------------
SUM: 42 7F 47 2D 48 76 95 07 43E9

C980 1C CB 1D 06 04 CB 1A CB : BE
C988 1C CB 1D 10 F8 22 25 C8 : 1B
C990 3A 2C C8 C6 40 32 2C C8 : 5A
C998 20 DE C1 C9 CD 0D CA 16 : 42
C9A0 00 CD BA C9 01 D0 1F ED : 2D
C9A8 78 F6 10 ED 79 CD BA C9 : 34
C9B0 01 D0 1F ED 78 E6 EF ED : 17
C9B8 79 C9 E5 CD C7 C9 01 00 : 85
C9C0 04 09 CD C7 C9 E1 C9 4D : 61
C9C8 44 ED 78 A3 28 04 37 CB : 7A
C9D0 12 C9 B7 CB 12 C9 2A 03 : 65
C9D8 C8 ED 5B 05 C8 AF 32 F6 : B4
C9E0 FB 06 1D ED 41 CD 07 59 : 79
C9E8 06 1E ED 41 22 27 C8 7A : DD
C9F0 32 2B C8 C9 2A 2E C8 11 : 1F
C9F8 00 04 CD 27 E0 ED 5B 07 : 27
---------------------------------
SUM: D9 FB 87 6D FA E4 4C 10 034B

CA00 C8 EB 29 19 ED 5B A4 E0 : C1
CA08 19 22 29 C8 C9 3A 2C C8 : 23
CA10 2A 27 C8 84 67 3A 2B C8 : 31
CA18 5F C9                   : 28
---------------------------------
SUM: 6A FD 1A 65 1D CF FB 70 BD3E
```

▶ラスト・ハルマゲドンは７枚組だったが…… (X1版T&Tについて)。
渡辺　光 (18) 北海道

リスト5 X68000→X1turboZソースリスト

```
0000                1  ;
0000                2  ;X68K G-DATA ->X1turboZ G-DATA
0000                3  ;                      Ver 1.1
0000                4  ;    FOR S-OS REDA
0000                5
B400                6   ORG $B400
B400                7
CD00 P              8  #CC       EQU $CD00
B400                9
FBF6 P             10  #SCRNM2 EQU $FBF6
5907 P             11  #GRAADR EQU $5907
D00C P             12  #DEVI   EQU $D00C
E012 P             13  #LDAHL  EQU $E012
E027 P             14  #MLTHD  EQU $E027
E08C P             15  #STOP   EQU $E08C
E09C P             16  #IOFG   EQU $E09C
E0A4 P             17  #BADR   EQU $E0A4
B400               18
B400 C3 35 B4      19   JP BEGIN
B403               20
B403               21  ;
B403 00 00         22  XF   DW 0
B405 00 00         23  YF   DW 0
B407 00 00         24  XT   DW 0
B409 00 00         25  YT   DW 0
B40B 00 00         26  DXF  DW 0
B40D 00 00         27  DYF  DW 0
B40F 00 00         28  DXT  DW 0
B411 00 00         29  DYT  DW 0
B413 00 00         30  EY   DW 0
B415               31
B415 00 00         32  XF1  DW 0
B417 00 00         33  YF1  DW 0
B419 00 02         34  XF2  DW 512
B41B 00 02         35  YF2  DW 512
B41D 00 00         36  XT1  DW 0
B41F 00 00         37  YT1  DW 0
B421 40 01         38  XT2  DW 320
B423 C8 00         39  YT2  DW 200
B425               40
B425 00 00         41  ADR  DW 0
B427 00 00         42  ADR2 DW 0
B429 00 00         43  YB   DW 0
B42B 00            44  DXT2 DB 0
B42C 00 00         45  DXT3 DW 0
B42E 00 00         46  RADR DW 0
B430 00 00         47  DTCC DW 0
B432 00            48  HFLG DB 0
B433 00            49  VFLG DB 0
B434 00            50  NF16 DB 0
B435               51  ;
B435               52
B435               53  BEGIN
B435 E1            54   POP HL
B436 22 2E B4      55   LD (RADR),HL
B439 2A 13 B4      56   LD HL,(EY)
B43C 7C            57   LD A,H
B43D B5            58   OR L
B43E C2 1C B5      59   JP NZ,ENTER
B441 21 00 00      60   LD HL,0
B444 22 29 B4      61   LD (YB),HL
B447 2A 19 B4      62   LD HL,(XF2)
B44A ED 5B 15 B4   63   LD DE,(XF1)
B44E B7            64   OR A
B44F ED 52         65   SBC HL,DE
B451 22 0B B4      66   LD (DXF),HL
B454 2A 1B B4      67   LD HL,(YF2)
B457 ED 5B 17 B4   68   LD DE,(YF1)
B45B B7            69   OR A
B45C ED 52         70   SBC HL,DE
B45E 22 0D B4      71   LD (DYF),HL
B461               72
B461 2A 21 B4      73   LD HL,(XT2)
B464 ED 5B 1D B4   74   LD DE,(XT1)
B468 B7            75   OR A
B469 ED 52         76   SBC HL,DE
B46B 22 0F B4      77   LD (DXT),HL
B46E B7            78   OR A
B46F CB 1C         79   RR H
B471 CB 1D         80   RR L
B473 CB 1C         81   RR H
B475 CB 1D         82   RR L
B477 CB 1C         83   RR H
B479 CB 1D         84   RR L
B47B 7D            85   LD A,L
B47C 32 2B B4      86   LD (DXT2),A
B47F               87
B47F 2A 23 B4      88   LD HL,(YT2)
B482 ED 5B 1F B4   89   LD DE,(YT1)
B486 B7            90   OR A
B487 ED 52         91   SBC HL,DE
B489 22 11 B4      92   LD (DYT),HL
B48C               93
B48C 2A 17 B4      94   LD HL,(YF1)
B48F 22 05 B4      95   LD (YF),HL
B492 2A 0D B4      96   LD HL,(DYF)
B495 29            97   ADD HL,HL
B496 ED 5B 11 B4   98   LD DE,(DYT)
B49A B7            99   OR A
B49B ED 52        100   SBC HL,DE
B49D 22 13 B4     101   LD (EY),HL
B4A0              102
B4A0 2A 1F B4     103   LD HL,(YT1)
B4A3 22 09 B4     104   LD (YT),HL
B4A6              105  YLP
B4A6 AF           106     XOR A
B4A7 32 33 B4     107     LD (VFLG),A
B4AA              108  WHILE2
B4AA 3A 14 B4     109     LD A,(EY+1)
B4AD FE 80        110     CP $80
B4AF D2 3A B5     111     JP NC,WEND2
B4B2 2A 15 B4     112       LD HL,(XF1)
B4B5 22 03 B4     113       LD (XF),HL
B4B8 2A 0B B4     114       LD HL,(DXF)
B4BB 29           115       ADD HL,HL
B4BC ED 5B 0F B4  116       LD DE,(DXT)
B4C0 B7           117       OR A
B4C1 ED 52        118       SBC HL,DE
B4C3 44           119       LD B,H   ;
B4C4 4D           120       LD C,L   ;BC=EX
B4C5 2A 1D B4     121       LD HL,(XT1)
B4C8 22 07 B4     122       LD (XT),HL
B4CB CD C7 B6     123       CALL XYADR2
B4CE              124  XLP
B4CE AF           125       XOR A
B4CF 32 32 B4     126       LD (HFLG),A
B4D2 2A 07 B4     127       LD HL,(XT)
B4D5 29           128       ADD HL,HL
B4D6 11 00 CD     129       LD DE,#CC
B4D9 19           130       ADD HL,DE
B4DA 22 30 B4     131       LD (DTCC),HL
B4DD              132  WHILE1
B4DD 3E 7F        133         LD A,$7F
B4DF B8           134         CP B
B4E0 38 12        135         JR C,WEND1
B4E2 CD 70 B5     136           CALL POINT
B4E5 2A 0F B4     137           LD HL,(DXT)
B4E8 29           138           ADD HL,HL
B4E9 50           139           LD D,B
B4EA 59           140           LD E,C
B4EB EB           141           EX DE,HL
B4EC B7           142           OR A
B4ED ED 52        143           SBC HL,DE
B4EF 44           144           LD B,H
B4F0 4D           145           LD C,L
B4F1 C3 DD B4     146           JP WHILE1
B4F4              147  WEND1
B4F4 2A 0B B4     148         LD HL,(DXF)
B4F7 29           149         ADD HL,HL
B4F8 09           150         ADD HL,BC
B4F9 44           151         LD B,H
B4FA 4D           152         LD C,L
B4FB 2A 07 B4     153       LD HL,(XT)
B4FE 23           154       INC HL
B4FF 22 07 B4     155       LD (XT),HL
B502 ED 5B 21 B4  156       LD DE,(XT2)
B506 B7           157       OR A
B507 ED 52        158       SBC HL,DE
B509 C2 CE B4     159       JP NZ,XLP
B50C              160
B50C 2A 29 B4     161       LD HL,(YB)
B50F 23           162       INC HL
B510 22 29 B4     163       LD (YB),HL
B513 11 04 00     164       LD DE,4
B516 B7           165       OR A
B517 ED 52        166       SBC HL,DE
B519 CA 65 B5     167       JP Z,EXIT
B51C              168  ENTER
B51C 2A 05 B4     169       LD HL,(YF)
B51F 23           170       INC HL
B520 22 05 B4     171       LD (YF),HL
B523 2A 11 B4     172       LD HL,(DYT)
B526 29           173       ADD HL,HL
B527 ED 5B 13 B4  174       LD DE,(EY)
B52B EB           175       EX DE,HL
B52C B7           176       OR A
B52D ED 52        177       SBC HL,DE
B52F 22 13 B4     178       LD (EY),HL
B532 3E 01        179       LD A,1
B534 32 33 B4     180       LD (VFLG),A
B537 C3 AA B4     181       JP WHILE2
B53A              182  WEND2
B53A 2A 0D B4     183     LD HL,(DYF)
B53D 29           184     ADD HL,HL
B53E ED 5B 13 B4  185     LD DE,(EY)
B542 19           186     ADD HL,DE
B543 22 13 B4     187     LD (EY),HL
B546 CD D4 B5     188     CALL PSET
B549 2A 09 B4     189     LD HL,(YT)
B54C 23           190     INC HL
B54D 22 09 B4     191     LD (YT),HL
B550 ED 5B 23 B4  192     LD DE,(YT2)
B554 B7           193     OR A
B555 ED 52        194     SBC HL,DE
B557 C2 A6 B4     195     JP NZ,YLP
B55A 21 00 00     196   LD HL,0
B55D 22 13 B4     197   LD (EY),HL
B560 2A 2E B4     198   LD HL,(RADR)
B563 E5           199   PUSH HL
B564 C9           200   RET
B565              201
B565              202  ;
B565              203  EXIT
B565 21 00 00     204   LD HL,0
B568 22 29 B4     205   LD (YB),HL
B56B 2A 2E B4     206   LD HL,(RADR)
B56E E5           207   PUSH HL
B56F C9           208   RET
B570              209
B570              210  ;
B570              211  ;X68K DATA(65536)
B570              212  ;
B570              213  ;OUT:CC(XT+0,1)
B570              214  POINT
B570 C5           215   PUSH BC
B571 DD 2A 30 B4  216   LD IX,(DTCC)
B575 2A 27 B4     217   LD HL,(ADR2)
B578 CD 12 E0     218   CALL #LDAHL
B57B 47           219   LD B,A
B57C 23           220   INC HL
B57D CD 12 E0     221   CALL #LDAHL
B580 4F           222   LD C,A
B581 23           223   INC HL
B582 22 27 B4     224   LD (ADR2),HL
B585 3A 33 B4     225   LD A,(VFLG)
B588 B7           226   OR A
B589 C2 A0 B5     227   JP NZ,PSKIP
B58C 3A 32 B4     228   LD A,(HFLG)
B58F B7           229   OR A
B590 C2 A0 B5     230   JP NZ,PSKIP
B593 DD 70 00     231     LD (IX+0),B
B596 DD 71 01     232     LD (IX+1),C
B599 3C           233     INC A
B59A 32 32 B4     234     LD (HFLG),A
B59D C3 D2 B5     235     JP OCOL
B5A0              236  PSKIP
B5A0              237
B5A0 DD 66 00     238   LD H,(IX+0) ;
B5A3 DD 6E 01     239   LD L,(IX+1) ;
B5A6 B7           240   OR A        ;
B5A7 ED 42        241   SBC HL,BC   ;
B5A9 28 27        242   JR Z,OCOL   ;
B5AB 09           243   ADD HL,BC   ;
B5AC EB           244   EX DE,HL    ;
B5AD FD 21 E0 B6  245   LD IY,YUSEN ;BC
B5B1 3E 03        246   LD A,3      ;A
B5B3              247  PRELP
B5B3 FD 6E 00     248     LD L,(IY+0) ;HL
B5B6 FD 66 01     249     LD H,(IY+1)
B5B9 B7           250     OR A
B5BA ED 52        251     SBC HL,DE
B5BC 28 14        252     JR Z,OCOL
B5BE 19           253     ADD HL,DE
B5BF B7           254     OR A
B5C0 ED 42        255     SBC HL,BC
B5C2 28 08        256     JR Z,NCOL
B5C4 FD 23        257     INC IY
B5C6 FD 23        258     INC IY
B5C8 3D           259   DEC A
B5C9 C2 B3 B5     260   JP NZ,PRELP
B5CC              261   ;
B5CC              262  NCOL
B5CC DD 70 00     263   LD (IX+0),B
B5CF DD 71 01     264   LD (IX+1),C
B5D2              265  OCOL
B5D2 C1           266   POP BC
B5D3 C9           267   RET
B5D4              268
B5D4              269
B5D4              270  ;FOR XT=XT1 TO XT2
B5D4              271  ;  PSET(XT,YT),CC(XT)
B5D4              272  ;NEXT
B5D4              273  PSET
B5D4 AF           274   XOR A
B5D5 32 34 B4     275   LD (NF16),A
B5D8 2A 1D B4     276   LD HL,(XT1)
B5DB 29           277   ADD HL,HL
B5DC 01 00 CD     278   LD BC,#CC
B5DF 09           279   ADD HL,BC    ;HL=CC+XT1*2
B5E0 22 2C B4     280   LD (DXT3),HL
B5E3 CD AD B6     281   CALL XYADR
B5E6              282
B5E6 3E C0        283   LD A,$C0
B5E8 CD FB B5     284   CALL PSET1
B5EB CD 3B B6     285   CALL RRBIT ;GREEN
B5EE 3E 80        286   LD A,$80
B5F0 CD FB B5     287   CALL PSET1
B5F3 CD 3B B6     288   CALL RRBIT ;RED
B5F6 3E 40        289   LD A,$40
B5F8 C3 FB B5     290   JP PSET1
B5FB              291
B5FB              292  ;
B5FB              293  ;GRAM 4 PLANE
B5FB              294  ;
B5FB              295  ;IN:CC(XT+0,1),A
B5FB              296  PSET1
B5FB 2A 25 B4     297   LD HL,(ADR)
B5FE 84           298   ADD A,H
B5FF 67           299   LD H,A
B600 CD 19 B6     300   CALL DTOBC2 ;BANK0
B603 01 D0 1F     301     LD BC,$1FD0
B606 ED 78        302     IN A,(C)
B608 F6 10        303     OR $10
B60A ED 79        304     OUT (C),A
B60C CD 19 B6     305   CALL DTOBC2 ;BANK1
B60F 01 D0 1F     306     LD BC,$1FD0
B612 ED 78        307     IN A,(C)
B614 E6 EF        308     AND $EF
B616 ED 79        309     OUT (C),A
B618 C9           310   RET
B619              311
B619              312  DTOBC2
B619 E5           313   PUSH HL
B61A CD 6B B6     314   CALL RRBIT2 ;$x000
B61D 3A 34 B4     315   LD A,(NF16)
B620 3C           316   INC A
B621 32 34 B4     317   LD (NF16),A
B624 FE 04        318   CP 4
B626 C2 32 B6     319   JP NZ,UP8
B629 ED 5B 2C B4  320     LD DE,(DXT3)
B62D 13           321     INC DE
B62E ED 53 2C B4  322     LD (DXT3),DE
B632              323  UP8
B632 7C           324   LD A,H
B633 C6 04        325   ADD A,4
B635 67           326   LD H,A
B636 CD 6B B6     327   CALL RRBIT2 ;$x400
B639 E1           328   POP HL
B63A C9           329   RET
B63B              330
B63B              331  ;
B63B              332  RRBIT
B63B DD 2A 2C B4  333   LD IX,(DXT3)
B63F 11 10 00     334   LD DE,16
B642 3A 2B B4     335   LD A,(DXT2)
B645 47           336   LD B,A
B646              337  RRBITLP
B646 DD CB 00 16  338     RL (IX+0)  ;
B64A DD CB 02 16  339     RL (IX+2)  ;
B64E DD CB 04 16  340     RL (IX+4)  ;
B652 DD CB 06 16  341     RL (IX+6)  ;
B656 DD CB 08 16  342     RL (IX+8)  ;
B65A DD CB 0A 16  343     RL (IX+10) ;
B65E DD CB 0C 16  344     RL (IX+12) ;
B662 DD CB 0E 16  345     RL (IX+14) ;
B666 DD 19        346     ADD IX,DE
B668 10 DC        347     DJNZ RRBITLP
B66A C9           348   RET
B66B              349
B66B              350  ;
B66B              351  RRBIT2
B66B E5           352   PUSH HL
B66C 44           353   LD B,H
B66D 4D           354   LD C,L
B66E DD 2A 2C B4  355   LD IX,(DXT3)
B672 11 10 00     356   LD DE,16
B675 3A 2B B4     357   LD A,(DXT2)
B678 67           358   LD H,A
B679              359  RRBITLP2
B679 AF           360     XOR A
B67A DD CB 00 16  361     RL (IX+0)
B67E 17           362     RLA
B67F DD CB 02 16  363     RL (IX+2)
B683 17           364     RLA
B684 DD CB 04 16  365     RL (IX+4)
B688 17           366     RLA
B689 DD CB 06 16  367     RL (IX+6)
B68D 17           368     RLA
B68E DD CB 08 16  369     RL (IX+8)
B692 17           370     RLA
B693 DD CB 0A 16  371     RL (IX+10)
B697 17           372     RLA
B698 DD CB 0C 16  373     RL (IX+12)
B69C 17           374     RLA
B69D DD CB 0E 16  375     RL (IX+14)
B6A1 17           376     RLA
B6A2 ED 79        377     OUT (C),A
B6A4 03           378     INC BC
B6A5 DD 19        379     ADD IX,DE
B6A7 25           380     DEC H
B6A8 C2 79 B6     381     JP NZ,RRBITLP2
B6AB E1           382   POP HL
B6AC C9           383   RET
B6AD              384
B6AD              385  ;FOR X1
B6AD              386  ;
B6AD              387  XYADR ;IN XT,YT OUT ADR,BIT
B6AD 2A 1D B4     388   LD HL,(XT1)
B6B0 ED 5B 09 B4  389   LD DE,(YT)
B6B4 AF           390   XOR A
B6B5 32 F6 FB     391   LD (#SCRNM2),A
B6B8 06 1D        392   LD B,$1D
B6BA ED 41        393   OUT (C),B
B6BC CD 07 59     394   CALL #GRAADR
B6BF 06 1E        395   LD B,$1E
B6C1 ED 41        396   OUT (C),B
B6C3 22 25 B4     397   LD (ADR),HL
B6C6 C9           398   RET
B6C7              399
B6C7              400  ;FOR X68K
B6C7              401  ;
B6C7              402  XYADR2 ;IN XF,YF OUT ADR2
B6C7 2A 29 B4     403   LD HL,(YB)
B6CA 11 00 04     404   LD DE,1024    ;512*2
B6CD CD 27 E0     405   CALL #MLTHD
B6D0 ED 5B 03 B4  406   LD DE,(XF)
B6D4 EB           407   EX DE,HL
B6D5 29           408   ADD HL,HL
B6D6 19           409   ADD HL,DE     ;Y*512*2+X*2
B6D7 ED 5B A4 E0  410   LD DE,(#BADR)
B6DB 19           411   ADD HL,DE
B6DC 22 27 B4     412   LD (ADR2),HL
B6DF C9           413   RET
B6E0              414
B6E0              415  ;
B6E0              416  ;PRW DATA
B6E0              417  ;
B6E0              418  YUSEN
B6E0 00 00 00 00  419   DW 0,0,0
B6E4 00 00
B6E6              420
```


▶鳥取は３日遅れなんです。Oh!Xの発売日。そんなことはどうでもいいのですが，Oh!Xのとなりに「P○AYBOY」とか「デラ．ぺっ○ん」などという本をおくのだけはやめてほしい。このあたりの本屋は，どこへ行ってもこのテの本が必ずとなりにおいてあるんですよ。
井関　雅晶（16）鳥取県

リスト6　X68000→X1turboソースリスト

```
0000                  1  ;
0000                  2  ;X68K G-DATA ->X1turbo G-DATA
0000                  3  ;  (65536)          (8)
0000                  4  ;                    Ver 1.0
0000                  5  ;    FOR S-OS REDA
0000                  6
B000                  7     ORG $B000
B000                  8
B800 P                9  #LCP     EQU $B800
C000 P               10  #LCN     EQU $C000
B000                 11
CB00 P               12  #CC      EQU $CB00
B000                 13
FBF6 P               14  #SCRNM2 EQU $FBF6
5907 P               15  #GRAADR EQU $5907
D00C P               16  #DEVI   EQU $D00C
E012 P               17  #LDAHL  EQU $E012
E027 P               18  #MLTHD  EQU $E027
E08C P               19  #STOP   EQU $E08C
E09C P               20  #IOFG   EQU $E09C
E0A4 P               21  #BADR   EQU $E0A4
B000                 22
B000 C3 36 B0        23   JP BEGIN
B003                 24
B003                 25  ;
B003 00 00           26  XF    DW 0
B005 00 00           27  YF    DW 0
B007 00 00           28  XT    DW 0
B009 00 00           29  YT    DW 0
B00B 00 00           30  DXF   DW 0
B00D 00 00           31  DYF   DW 0
B00F 00 00           32  DXT   DW 0
B011 00 00           33  DYT   DW 0
B013 00 00           34  EY    DW 0
B015                 35
B015 00 00           36  XF1   DW 0
B017 00 00           37  YF1   DW 0
B019 00 02           38  XF2   DW 512
B01B 00 02           39  YF2   DW 512
B01D 00 00           40  XT1   DW 0
B01F 00 00           41  YT1   DW 0
B021 80 02           42  XT2   DW 640
B023 90 01           43  YT2   DW 400
B025                 44
B025 00              45  HMF   DB 0 ;0..400 1..200
B026                 46
B026 00 00           47  ADR   DW 0
B028 00 00           48  ADR2  DW 0
B02A 00 00           49  YB    DW 0
B02C 00              50  DXT2  DB 0
B02D 00 00           51  DXT3  DW 0
B02F 00 00           52  RADR  DW 0
B031 00 00           53  DTCC  DW 0
B033 00              54  HFLG  DB 0
B034 00              55  VFLG  DB 0
B035 00              56  SFLG  DB 0
B036                 57  ;
B036                 58
B036                 59  BEGIN
B036 E1              60   POP HL
B037 22 2F B0        61   LD (RADR),HL
B03A 3A 35 B0        62   LD A,(SFLG)
B03D B7              63   OR A
B03E C2 38 B1        64   JP NZ,ENTER
B041                 65
B041 3E 01           66   LD A,1
B043 32 35 B0        67   LD (SFLG),A
B046 21 00 B8        68   LD HL,#LCP
B049 11 01 B8        69   LD DE,#LCP+1
B04C 01 FF 0F        70   LD BC,$FFF
B04F AF              71   XOR A
B050 77              72   LD (HL),A
B051 ED B0           73   LDIR
B053                 74
B053 21 00 00        75   LD HL,0
B056 22 2A B0        76   LD (YB),HL
B059 2A 19 B0        77   LD HL,(XF2)
B05C ED 5B 15 B0     78   LD DE,(XF1)
B060 B7              79   OR A
B061 ED 52           80   SBC HL,DE
B063 22 0B B0        81   LD (DXF),HL
B066 2A 1B B0        82   LD HL,(YF2)
B069 ED 5B 17 B0     83   LD DE,(YF1)
B06D B7              84   OR A
B06E ED 52           85   SBC HL,DE
B070 22 0D B0        86   LD (DYF),HL
B073                 87
B073 2A 21 B0        88   LD HL,(XT2)
B076 ED 5B 1D B0     89   LD DE,(XT1)
B07A B7              90   OR A
B07B ED 52           91   SBC HL,DE
B07D 22 0F B0        92   LD (DXT),HL
B080 B7              93   OR A
B081 CB 1C           94   RR H
B083 CB 1D           95   RR L
B085 CB 1C           96   RR H
B087 CB 1D           97   RR L
B089 CB 1C           98   RR H
B08B CB 1D           99   RR L
B08D 7D             100   LD A,L
B08E 32 2C B0       101   LD (DXT2),A
B091                102
B091 2A 23 B0       103   LD HL,(YT2)
B094 ED 5B 1F B0    104   LD DE,(YT1)
B098 B7             105   OR A
B099 ED 52          106   SBC HL,DE
B09B 22 11 B0       107   LD (DYT),HL
B09E                108
B09E 2A 17 B0       109   LD HL,(YF1)
B0A1 22 05 B0       110   LD (YF),HL
B0A4 2A 0D B0       111   LD HL,(DYF)
B0A7 29             112   ADD HL,HL
B0A8 ED 5B 11 B0    113   LD DE,(DYT)
B0AC B7             114   OR A
B0AD ED 52          115   SBC HL,DE
B0AF 22 13 B0       116   LD (EY),HL
B0B2                117
B0B2 2A 1F B0       118   LD HL,(YT1)
B0B5 22 09 B0       119   LD (YT),HL
B0B8                120  YLP
B0B8 AF             121      XOR A
B0B9 32 34 B0       122      LD (VFLG),A
B0BC                123  WHILE2
B0BC 3A 14 B0       124      LD A,(EY+1)
B0BF FE 80          125      CP $80
B0C1 D2 56 B1       126      JP NC,WEND2
B0C4 2A 15 B0       127       LD HL,(XF1)
B0C7 22 03 B0       128       LD (XF),HL
B0CA 2A 0B B0       129       LD HL,(DXF)
B0CD 29             130       ADD HL,HL
B0CE ED 5B 0F B0    131       LD DE,(DXT)
B0D2 B7             132       OR A
B0D3 ED 52          133       SBC HL,DE
B0D5 44             134       LD B,H  ;
B0D6 4D             135       LD C,L  ;BC=EX
B0D7 2A 1D B0       136       LD HL,(XT1)
B0DA 22 07 B0       137       LD (XT),HL
B0DD CD 05 B3       138       CALL XYADR2
B0E0                139  XLP
B0E0 AF             140        XOR A
B0E1 32 33 B0       141        LD (HFLG),A
B0E4 2A 07 B0       142        LD HL,(XT)
B0E7 29             143        ADD HL,HL
B0E8 11 00 CB       144        LD DE,#CC
B0EB 19             145        ADD HL,DE
B0EC 22 31 B0       146        LD (DTCC),HL
B0EF                147  WHILE1
B0EF 3E 7F          148          LD A,$7F
B0F1 B8             149          CP B
B0F2 38 12          150          JR C,WEND1
B0F4 CD 9D B1       151           CALL POINT
B0F7 2A 0F B0       152           LD HL,(DXT)
B0FA 29             153           ADD HL,HL
B0FB 50             154           LD D,B
B0FC 59             155           LD E,C
B0FD EB             156           EX DE,HL
B0FE B7             157           OR A
B0FF ED 52          158           SBC HL,DE
B101 44             159           LD B,H
B102 4D             160           LD C,L
B103 C3 EF B0       161           JP WHILE1
B106                162  WEND1
B106 2A 0B B0       163          LD HL,(DXF)
B109 29             164          ADD HL,HL
B10A 09             165          ADD HL,BC
B10B 44             166          LD B,H
B10C 4D             167          LD C,L
B10D 3A 33 B0       168        LD A,(HFLG)
B110 21 34 B0       169        LD HL,VFLG
B113 B6             170        OR (HL)
B114 CC 8C B1       171        CALL Z,REF
B117 2A 07 B0       172        LD HL,(XT)
B11A 23             173        INC HL
B11B 22 07 B0       174        LD (XT),HL
B11E ED 5B 21 B0    175        LD DE,(XT2)
B122 B7             176        OR A
B123 ED 52          177        SBC HL,DE
B125 DA E0 B0       178        JP C,XLP
B128                179
B128 2A 2A B0       180        LD HL,(YB)
B12B 23             181        INC HL
B12C 22 2A B0       182        LD (YB),HL
B12F 11 04 00       183        LD DE,4
B132 B7             184        OR A
B133 ED 52          185        SBC HL,DE
B135 CA 81 B1       186        JP Z,EXIT
B138                187  ENTER
B138 2A 05 B0       188        LD HL,(YF)
B13B 23             189        INC HL
B13C 22 05 B0       190        LD (YF),HL
B13F 2A 11 B0       191        LD HL,(DYT)
B142 29             192        ADD HL,HL
B143 ED 5B 13 B0    193        LD DE,(EY)
B147 EB             194        EX DE,HL
B148 B7             195        OR A
B149 ED 52          196        SBC HL,DE
B14B 22 13 B0       197        LD (EY),HL
B14E 3E 01          198        LD A,1
B150 32 34 B0       199        LD (VFLG),A
B153 C3 BC B0       200        JP WHILE2
B156                201  WEND2
B156 2A 0D B0       202     LD HL,(DYF)
B159 29             203     ADD HL,HL
B15A ED 5B 13 B0    204     LD DE,(EY)
B15E 19             205     ADD HL,DE
B15F 22 13 B0       206     LD (EY),HL
B162 CD FB B1       207     CALL PSET
B165 2A 09 B0       208     LD HL,(YT)
B168 23             209     INC HL
B169 22 09 B0       210     LD (YT),HL
B16C ED 5B 23 B0    211     LD DE,(YT2)
B170 B7             212     OR A
B171 ED 52          213     SBC HL,DE
B173 DA B8 B0       214     JP C,YLP
B176 21 00 00       215   LD HL,0
B179 22 13 B0       216   LD (EY),HL
B17C 2A 2F B0       217   LD HL,(RADR)
B17F E5             218   PUSH HL
B180 C9             219   RET
B181                220
B181                221  ;
B181                222  EXIT
B181 21 00 00       223   LD HL,0
B184 22 2A B0       224   LD (YB),HL
B187 2A 2F B0       225   LD HL,(RADR)
B18A E5             226   PUSH HL
B18B C9             227   RET
B18C                228
B18C                229  ;
B18C                230  REF
B18C DD 2A 31 B0    231   LD IX,(DTCC)
B190 DD 7E FE       232   LD A,(IX-2)
B193 DD 77 00       233   LD (IX+0),A
B196 DD 7E FF       234   LD A,(IX-1)
B199 DD 77 01       235   LD (IX+1),A
B19C C9             236   RET
B19D                237
B19D                238  ;
B19D                239  ;X68K DATA(65536)
B19D                240  ;
B19D                241  ;IN:CC(XT+0,1)
B19D                242  POINT
B19D C5             243   PUSH BC
B19E DD 2A 31 B0    244   LD IX,(DTCC)
B1A2 2A 28 B0       245   LD HL,(ADR2)
B1A5 CD 12 E0       246   CALL #LDAHL
B1A8 47             247   LD B,A
B1A9 23             248   INC HL
B1AA CD 12 E0       249   CALL #LDAHL
B1AD 4F             250   LD C,A
B1AE 23             251   INC HL
B1AF 22 28 B0       252   LD (ADR2),HL
B1B2 3A 34 B0       253   LD A,(VFLG)
B1B5 B7             254   OR A
B1B6 C2 C7 B1       255   JP NZ,PSKIP
B1B9 3A 33 B0       256   LD A,(HFLG)
B1BC B7             257   OR A
B1BD C2 C7 B1       258   JP NZ,PSKIP
B1C0 3C             259     INC A
B1C1 32 33 B0       260     LD (HFLG),A
B1C4 C3 F3 B1       261     JP NCOL
B1C7                262  PSKIP
B1C7                263
B1C7 DD 66 00       264   LD H,(IX+0)  ;DE=ALREADY
B1CA DD 6E 01       265   LD L,(IX+1)  ;    COLOR
B1CD B7             266   OR A         ;
B1CE ED 42          267   SBC HL,BC    ;
B1D0 28 27          268   JR Z,OCOL    ;
B1D2 09             269   ADD HL,BC    ;
B1D3 EB             270   EX DE,HL     ;
B1D4 FD 21 1E B3    271   LD IY,YUSEN  ;BC=POINT COLOR
B1D8 3E 03          272   LD A,3       ;A=NUMBER
B1DA                273  PRELP
B1DA FD 6E 00       274     LD L,(IY+0)
B1DD FD 66 01       275     LD H,(IY+1)
B1E0 B7             276     OR A
B1E1 ED 52          277     SBC HL,DE
B1E3 28 14          278     JR Z,OCOL
B1E5 19             279     ADD HL,DE
B1E6 B7             280     OR A
B1E7 ED 42          281     SBC HL,BC
B1E9 28 08          282     JR Z,NCOL
B1EB FD 23          283     INC IY
B1ED FD 23          284     INC IY
B1EF 3D             285   DEC A
B1F0 C2 DA B1       286   JP NZ,PRELP
B1F3                287   ;PRW COLOR
B1F3                288  NCOL
B1F3 DD 70 00       289   LD (IX+0),B
B1F6 DD 71 01       290   LD (IX+1),C
B1F9                291  OCOL
B1F9 C1             292   POP BC
B1FA C9             293   RET
B1FB                294
B1FB                295
B1FB                296  ;FOR XT=XT1 TO XT2
B1FB                297  ;  PSET(XT,YT),CC(XT)
B1FB                298  ;NEXT
B1FB                299  PSET
B1FB 2A 1D B0       300   LD HL,(XT1)
B1FE 29             301   ADD HL,HL
B1FF 01 00 CB       302   LD BC,#CC
B202 09             303   ADD HL,BC    ;HL=#CC+XT1*2
B203 22 2D B0       304   LD (DXT3),HL
B206 CD EB B2       305   CALL XYADR
B209                306
B209 FD 21 01 B8    307   LD IY,#LCP+1
B20D DD 21 01 C0    308   LD IX,#LCN+1
B211 ED 5B 1D B0    309   LD DE,(XT1)
B215 DD 19          310   ADD IX,DE
B217 FD 19          311   ADD IY,DE
B219 01 D0 1F       312    LD BC,$1FD0
B21C ED 78          313    IN A,(C)
B21E E6 EF          314    AND $EF
B220 ED 79          315    OUT (C),A
B222 3A 09 B0       316   LD A,(YT)
B225 E6 01          317   AND 1
B227 28 14          318   JR Z,EXIXIY
B229 DD E5          319    PUSH IX
B22B FD E5          320    PUSH IY
B22D DD E1          321    POP IX
B22F FD E1          322    POP IY
B231 3A 25 B0       323      LD A,(HMF)
B234 B7             324      OR A
B235 20 06          325      JR NZ,EXIXIY
B237 ED 78          326    IN A,(C)
B239 F6 10          327    OR $10
B23B ED 79          328    OUT (C),A
B23D                329  EXIXIY
B23D                330
B23D 3E C0          331   LD A,$C0
B23F CD 4C B2       332   CALL PSET1
B242 3E 80          333   LD A,$80
B244 CD 4C B2       334   CALL PSET1
B247 3E 40          335   LD A,$40
B249 C3 4C B2       336   JP PSET1
B24C                337
B24C                338  ;
B24C                339  ;GRAM 1 PLANE
B24C                340  ;
B24C                341  ;OUT:CC(XT+0,1),A
B24C                342  PSET1
B24C ED 4B 26 B0    343   LD BC,(ADR)
B250 80             344   ADD A,B
B251 47             345   LD B,A
B252 2A 2D B0       346   LD HL,(DXT3)
B255 3A 2C B0       347   LD A,(DXT2)
B258 5F             348   LD E,A
B259 DD E5          349   PUSH IX
B25B FD E5          350   PUSH IY
B25D                351  RRBITLP2
B25D D5             352    PUSH DE
B25E 16 08          353     LD D,8
B260 1E 00          354     LD E,0
B262                355  RRBITLP3
B262 CD 86 B2       356       CALL BIT5
B265 CD AF B2       357       CALL LCINC
B268 CB 13          358       RL E
B26A DD 23          359       INC IX
B26C FD 23          360       INC IY
B26E 15             361       DEC D
B26F C2 62 B2       362       JP NZ,RRBITLP3
B272 ED 59          363    OUT (C),E
B274 03             364    INC BC
B275 D1             365    POP DE
B276 1D             366    DEC E
B277 C2 5D B2       367    JP NZ,RRBITLP2
B27A FD E1          368   POP IY
B27C DD E1          369   POP IX
B27E 11 82 02       370   LD DE,642
B281 FD 19          371   ADD IY,DE
B283 DD 19          372   ADD IX,DE
B285 C9             373   RET
B286                374
B286                375  ;
B286                376  BIT5
B286 78             377   LD A,B
B287 FE 80          378   CP $80
B289 38 1C          379   JR C,BBIT5
B28B FE C0          380   CP $C0
B28D 38 06          381   JR C,RBIT5
B28F                382
B28F                383  GBIT5
B28F 7E             384   LD A,(HL)
B290 23             385   INC HL
B291 23             386   INC HL
B292 E6 F8          387   AND $F8 ;&Bxxxxx000
B294 C9             388   RET
B295                389
B295                390  RBIT5
B295 C5             391   PUSH BC
B296 46             392   LD B,(HL)
B297 23             393   INC HL
B298 7E             394   LD A,(HL)
B299 23             395   INC HL
B29A CB 18          396   RR B
B29C 1F             397   RRA
B29D CB 18          398   RR B
B29F 1F             399   RRA
B2A0 CB 18          400   RR B
B2A2 1F             401   RRA
B2A3 E6 F8          402   AND $F8
B2A5 C1             403   POP BC
B2A6 C9             404   RET
B2A7                405
B2A7                406  BBIT5
B2A7 23             407   INC HL
B2A8 7E             408   LD A,(HL)
B2A9 23             409   INC HL
B2AA E6 3E          410   AND $3E ;&B00111110
B2AC 07             411   RLCA
B2AD 07             412   RLCA
B2AE C9             413   RET
B2AF                414
```

▶夏休み終了2日前にして，宿題がほとんど終わっていない！　すでにあきらめた僕は「どの宿題を終わらせて，どの宿題を先生に頼んで期限を伸ばしてもらうか」の選別に入った。これを書いている間にも時間は刻々と減っていく。これというのも三国志IIをやりすぎたせいだ！　うおー，孔明よオレの宿題を手伝ってくれ～。　山野辺　太郎（14）宮城県

```
B2AF                415   ;
B2AF                416   ;
B2AF                417   LCINC ;IN A OUT cy
B2AF FD 86 00       418    ADD A,(IY+0)
B2B2 38 05          419    JR C,PIXON
B2B4 FE F8          420    CP $F8 ;&Bxxxxx000
B2B6 3F             421    CCF
B2B7 30 03          422    JR NC,PIXOFF
B2B9                423   PIXON
B2B9 D6 F8          424    SUB $F8
B2BB 37             425    SCF
B2BC                426   PIXOFF
B2BC F5             427    PUSH AF
B2BD CD C2 B2       428    CALL LCINC2
B2C0 F1             429    POP AF
B2C1 C9             430    RET
B2C2                431
B2C2                432   ;
B2C2                433   LCINC2 ;IN A
B2C2 E5             434    PUSH HL
B2C3 67             435    LD H,A
B2C4 CB 3F          436    SRL A          ;
B2C6 CB 3F          437    SRL A          ;A/8
B2C8 CB 3F          438    SRL A          ;
B2CA 6F             439    LD L,A
B2CB DD 77 01       440    LD (IX+1),A    ;RIGHT DOWN
B2CE                441
B2CE 7C             442    LD A,H         ;DOWN
B2CF E6 07          443    AND 7          ;
B2D1 DD 86 00       444    ADD A,(IX+0) ;LC(R,N,X)
B2D4 85             445    ADD A,L      ;
B2D5 DD 77 00       446    LD (IX+0),A  ;
B2D8                447
B2D8 7D             448    LD A,L       ;LEFT-DOWN
B2D9 87             449    ADD A,A      ;  A*2
B2DA DD 86 FF       450    ADD A,(IX-1) ;LC(R,N,X-1)
B2DD DD 77 FF       451    LD (IX-1),A  ;
B2E0                452
B2E0 7D             453    LD A,L       ;RIGHT
B2E1 87             454    ADD A,A      ;  A*4
B2E2 87             455    ADD A,A      ;
B2E3 FD 86 01       456    ADD A,(IY+1) ;
B2E6 FD 77 01       457    LD (IY+1),A  ;
B2E9 E1             458    POP HL
B2EA C9             459    RET
B2EB                460
B2EB                461   ;FOR X1
B2EB                462   ;
B2EB                463   XYADR ;IN XT,YT OUT ADR,BIT
B2EB 2A 1D B0       464    LD HL,(XT1)
B2EE ED 5B 09 B0    465    LD DE,(YT)
B2F2 AF             466    XOR A
B2F3 32 F6 FB       467    LD (#SCRNM2),A
B2F6 06 1D          468    LD B,$1D
B2F8 ED 41          469    OUT (C),B
B2FA CD 07 59       470    CALL #GRAADR
B2FD 06 1E          471    LD B,$1E
B2FF ED 41          472    OUT (C),B
B301 22 26 B0       473    LD (ADR),HL
B304 C9             474    RET
B305                475
B305                476   ;FOR X68K
B305                477   ;
B305                478   XYADR2 ;IN XF,YF OUT ADR2
B305 2A 2A B0       479    LD HL,(YB)
B308 11 00 04       480    LD DE,1024   ;512*2
B30B CD 27 E0       481    CALL #MLTHD
B30E ED 5B 03 B0    482    LD DE,(XF)
B312 EB             483    EX DE,HL
B313 29             484    ADD HL,HL
B314 19             485    ADD HL,DE    ;Y*512*2+X*2
B315 ED 5B A4 E0    486    LD DE,(#BADR)
B319 19             487    ADD HL,DE
B31A 22 28 B0       488    LD (ADR2),HL
B31D C9             489    RET
B31E                490
B31E                491   ;
B31E                492   ;PRW DATA
B31E                493   ;
B31E                494   YUSEN
B31E 00 00 00 00    495    DW 0,0,0
B322 00 00
B324                496
```

リスト7　X1turboZ→X68000ソースリスト

```
0000                  1   ;
0000                  2   ;X1turboZ G-DATA->X68K G-DATA
0000                  3   ;
0000                  4   ;    FOR S-OS REDA
0000                  5
C800                  6    ORG $C800
C800                  7
FBF6 P                8   #SCRNM2 EQU $FBF6
5907 P                9   #GRAADR EQU $5907
D00C P               10   #DEVI   EQU $D00C
E018 P               11   #LDDEA  EQU $E018
E027 P               12   #MLTHD  EQU $E027
E08C P               13   #STOP   EQU $E08C
E09C P               14   #IOFG   EQU $E09C
E0A4 P               15   #BADR   EQU $E0A4
C800                 16
C800 C3 32 C8        17    JP BEGIN
C803                 18
C803                 19   ;
C803 00 00           20   XF   DW 0
C805 00 00           21   YF   DW 0
C807 00 00           22   XT   DW 0
C809 00 00           23   YT   DW 0
C80B 00 00           24   DXF  DW 0
C80D 00 00           25   DYF  DW 0
C80F 00 00           26   DXT  DW 0
C811 00 00           27   DYT  DW 0
C813 00 00           28   EY   DW 0
C815                 29
C815 00 00           30   XF1  DW 0
C817 00 00           31   YF1  DW 0
C819 40 01           32   XF2  DW 320
C81B C8 00           33   YF2  DW 200
C81D 00 00           34   XT1  DW 0
C81F 00 00           35   YT1  DW 0
C821 00 02           36   XT2  DW 512
C823 00 02           37   YT2  DW 512
C825                 38
C825 00 00           39   RGBX DW 0    ;68 1DOT
C827 00 00           40   ADR  DW 0
C829 00 00           41   ADR2 DW 0
C82B 00              42   BIT  DB 0
C82C 00 00           43   RGBP DW 0
C82E 00 00           44   YB   DW 0
C830 00 00           45   RADR DW 0
C832                 46   ;
C832                 47
C832                 48   BEGIN
C832 E1              49    POP HL
C833 22 30 C8        50    LD (RADR),HL
C836 2A 13 C8        51    LD HL,(EY)
C839 7C              52    LD A,H
C83A B5              53    OR L
C83B C2 2B C9        54    JP NZ,ENTER
C83E 21 00 00        55    LD HL,0
C841 22 2E C8        56    LD (YB),HL
C844 2A 19 C8        57    LD HL,(XF2)
C847 ED 5B 15 C8     58    LD DE,(XF1)
C84B B7              59    OR A
C84C ED 52           60    SBC HL,DE
C84E 22 0B C8        61    LD (DXF),HL
C851 2A 1B C8        62    LD HL,(YF2)
C854 ED 5B 17 C8     63    LD DE,(YF1)
C858 B7              64    OR A
C859 ED 52           65    SBC HL,DE
C85B 22 0D C8        66    LD (DYF),HL
C85E                 67
C85E 2A 21 C8        68    LD HL,(XT2)
C861 ED 5B 1D C8     69    LD DE,(XT1)
C865 B7              70    OR A
C866 ED 52           71    SBC HL,DE
C868 22 0F C8        72    LD (DXT),HL
C86B 2A 23 C8        73    LD HL,(YT2)
C86E ED 5B 1F C8     74    LD DE,(YT1)
C872 B7              75    OR A
C873 ED 52           76    SBC HL,DE
C875 22 11 C8        77    LD (DYT),HL
C878                 78
C878 2A 17 C8        79    LD HL,(YF1)
C87B 22 05 C8        80    LD (YF),HL
C87E 2A 0D C8        81    LD HL,(DYF)
C881 29              82    ADD HL,HL
C882 ED 5B 11 C8     83    LD DE,(DYT)
C886 B7              84    OR A
C887 ED 52           85    SBC HL,DE
C889 22 13 C8        86    LD (EY),HL
C88C                 87
C88C 2A 1F C8        88    LD HL,(YT1)
C88F 22 09 C8        89    LD (YT),HL
C892                 90   YLP
C892 2A 15 C8        91          LD HL,(XF1)
C895 22 03 C8        92          LD (XF),HL
C898 2A 0B C8        93          LD HL,(DXF)
C89B 29              94          ADD HL,HL
C89C ED 5B 0F C8     95          LD DE,(DXT)
C8A0 B7              96          OR A
C8A1 ED 52           97          SBC HL,DE
C8A3 44              98          LD B,H   ;
C8A4 4D              99          LD C,L   ;BC=EX
C8A5 2A 1D C8       100          LD HL,(XT1)
C8A8 22 07 C8       101          LD (XT),HL
C8AB CD F4 C9       102          CALL XYADR2
C8AE                103   XLP
C8AE CD 69 C9       104             CALL POINT
C8B1 CD 52 C9       105             CALL PSET
C8B4                106   WHILE1
C8B4 21 00 80       107             LD HL,$8000
C8B7 B7             108             OR A
C8B8 ED 42          109             SBC HL,BC
C8BA 38 16          110             JR C,WEND1
C8BC 2A 03 C8       111                LD HL,(XF)
C8BF 23             112                INC HL
C8C0 22 03 C8       113                LD (XF),HL
C8C3 2A 0F C8       114                LD HL,(DXT)
C8C6 29             115                ADD HL,HL
C8C7 50             116                LD D,B
C8C8 59             117                LD E,C
C8C9 EB             118                EX DE,HL
C8CA B7             119                OR A
C8CB ED 52          120                SBC HL,DE
C8CD 44             121                LD B,H
C8CE 4D             122                LD C,L
C8CF C3 B4 C8       123                JP WHILE1
C8D2                124   WEND1
C8D2 2A 0B C8       125             LD HL,(DXF)
C8D5 29             126             ADD HL,HL
C8D6 09             127             ADD HL,BC
C8D7 44             128             LD B,H
C8D8 4D             129             LD C,L
C8D9 2A 07 C8       130          LD HL,(XT)
C8DC 23             131          INC HL
C8DD 22 07 C8       132          LD (XT),HL
C8E0 ED 5B 21 C8    133          LD DE,(XT2)
C8E4 B7             134          OR A
C8E5 ED 52          135          SBC HL,DE
C8E7 C2 AE C8       136          JP NZ,XLP
C8EA                137
C8EA                138   WHILE2
C8EA 2A 13 C8       139       LD HL,(EY)
C8ED 11 00 80       140       LD DE,$8000
C8F0 B7             141       OR A
C8F1 ED 52          142       SBC HL,DE
C8F3 D2 0F C9       143       JP NC,WEND2
C8F6 2A 05 C8       144          LD HL,(YF)
C8F9 23             145          INC HL
C8FA 22 05 C8       146          LD (YF),HL
C8FD 2A 11 C8       147          LD HL,(DYT)
C900 29             148          ADD HL,HL
C901 ED 5B 13 C8    149          LD DE,(EY)
C905 EB             150          EX DE,HL
C906 B7             151          OR A
C907 ED 52          152          SBC HL,DE
C909 22 13 C8       153          LD (EY),HL
C90C C3 EA C8       154          JP WHILE2
C90F                155   WEND2
C90F 2A 0D C8       156       LD HL,(DYF)
C912 29             157       ADD HL,HL
C913 ED 5B 13 C8    158       LD DE,(EY)
C917 19             159       ADD HL,DE
C918 22 13 C8       160       LD (EY),HL
C91B 2A 2E C8       161          LD HL,(YB)
C91E 23             162          INC HL
C91F 22 2E C8       163          LD (YB),HL
C922 11 04 00       164          LD DE,4
C925 B7             165          OR A
C926 ED 52          166          SBC HL,DE
C928 CA 47 C9       167          JP Z,EXIT
C92B                168   ENTER
C92B 2A 09 C8       169       LD HL,(YT)
C92E 23             170       INC HL
C92F 22 09 C8       171       LD (YT),HL
C932 ED 5B 23 C8    172       LD DE,(YT2)
C936 B7             173       OR A
C937 ED 52          174       SBC HL,DE
C939 C2 92 C8       175       JP NZ,YLP
C93C 21 00 00       176    LD HL,0
C93F 22 13 C8       177    LD (EY),HL
C942 2A 30 C8       178    LD HL,(RADR)
C945 E5             179    PUSH HL
C946 C9             180    RET
C947                181
C947                182   ;
C947                183   EXIT
C947 21 00 00       184    LD HL,0
C94A 22 2E C8       185    LD (YB),HL
C94D 2A 30 C8       186    LD HL,(RADR)
C950 E5             187    PUSH HL
C951 C9             188    RET
C952                189
C952                190   ;
C952                191   ;PSET(XT,YT,(RGBX))
C952                192   ;
C952                193   PSET
C952 ED 5B 29 C8    194    LD DE,(ADR2)
C956 3A 26 C8       195    LD A,(RGBX+1)
C959 CD 18 E0       196    CALL #LDDEA
C95C 13             197    INC DE
C95D 3A 25 C8       198    LD A,(RGBX)
C960 CD 18 E0       199    CALL #LDDEA
C963 13             200    INC DE
C964 ED 53 29 C8    201    LD (ADR2),DE
C968 C9             202    RET
C969                203
C969                204   ;
C969                205   ;POINT(XF,YF)->X68K DATA
C969                206   ;
C969                207   ;OUT:(RGBX)
C969                208   POINT
C969 C5             209    PUSH BC
C96A CD D6 C9       210    CALL XYADR
C96D 21 00 00       211    LD HL,0
C970 22 25 C8       212    LD (RGBX),HL
C973 3E 40          213    LD A,$40
C975 32 2C C8       214    LD (RGBP),A
C978                215   PLPX1
C978 CD 9C C9       216       CALL POINT1
C97B 2A 25 C8       217       LD HL,(RGBX)
C97E B7             218       OR A
C97F CB 1C          219       RR H
C981 CB 1D          220       RR L
C983 06 04          221       LD B,4
C985                222   PLPRL
C985 CB 1A          223          RR D
C987 CB 1C          224          RR H
C989 CB 1D          225          RR L
C98B 10 F8          226          DJNZ PLPRL
C98D 22 25 C8       227       LD (RGBX),HL
C990 3A 2C C8       228       LD A,(RGBP)
C993 C6 40          229       ADD A,$40
C995 32 2C C8       230       LD (RGBP),A
C998 20 DE          231    JR NZ,PLPX1
C99A C1             232    POP BC
C99B C9             233    RET
C99C                234
C99C                235   ;GRAM 4 PLANE
C99C                236   ;  (AT 4096 COLOR)
C99C                237   ;OUT:D
C99C                238   POINT1
C99C CD 0D CA       239    CALL SETHLE   ;HL=(ADR),E=BIT
C99F 16 00          240    LD D,0
C9A1 CD BA C9       241    CALL BCTOD2   ;BANK 1
C9A4 01 D0 1F       242       LD BC,$1FD0
C9A7 ED 78          243       IN A,(C)
C9A9 F6 10          244       OR $10
C9AB ED 79          245       OUT (C),A
C9AD CD BA C9       246    CALL BCTOD2   ;BANK 2
C9B0 01 D0 1F       247       LD BC,$1FD0
C9B3 ED 78          248       IN A,(C)
C9B5 E6 EF          249       AND $EF
C9B7 ED 79          250       OUT (C),A
C9B9 C9             251    RET           ;D=&B0000xxxx
C9BA                252
C9BA                253   BCTOD2         ;$x000 $x400
C9BA E5             254    PUSH HL       ;
C9BB CD C7 C9       255    CALL BCTOD    ;GRAM 2 PLANE
C9BE 01 00 04       256    LD BC,$400    ;
C9C1 09             257    ADD HL,BC
C9C2 CD C7 C9       258    CALL BCTOD
C9C5 E1             259    POP HL
C9C6 C9             260    RET
C9C7                261
C9C7                262   BCTOD          ;D REG. 0 BIT
C9C7 4D             263    LD C,L        ;  ON or OFF
C9C8 44             264    LD B,H        ;(IF MASK(E)=ON)
C9C9 ED 78          265    IN A,(C)      ;
C9CB A3             266    AND E         ;GRAM 1 PLANE
C9CC 28 04          267    JR Z,NOBIT    ;
C9CE 37             268    SCF
C9CF CB 12          269    RL D
C9D1 C9             270    RET
C9D2                271   NOBIT
C9D2 B7             272    OR A
C9D3 CB 12          273    RL D
C9D5 C9             274    RET
C9D6                275
C9D6                276   ;FOR X1
C9D6                277   ;
C9D6                278   XYADR ;IN XF,YF OUT ADR,BIT
C9D6 2A 03 C8       279    LD HL,(XF)
C9D9 ED 5B 05 C8    280    LD DE,(YF)
C9DD AF             281    XOR A
C9DE 32 F6 FB       282    LD (#SCRNM2),A
C9E1 06 1D          283    LD B,$1D
C9E3 ED 41          284    OUT (C),B
C9E5 CD 07 59       285    CALL #GRAADR
C9E8 06 1E          286    LD B,$1E
C9EA ED 41          287    OUT (C),B
C9EC 22 27 C8       288    LD (ADR),HL
C9EF 7A             289    LD A,D
C9F0 32 2B C8       290    LD (BIT),A
C9F3 C9             291    RET
C9F4                292
C9F4                293   ;FOR X68K
C9F4                294   ;
C9F4                295   XYADR2 ;IN XT,YB OUT ADR2
C9F4 2A 2E C8       296    LD HL,(YB)
C9F7 11 00 04       297    LD DE,1024   ;512*2
C9FA CD 27 E0       298    CALL #MLTHD
C9FD ED 5B 07 C8    299    LD DE,(XT)
CA01 EB             300    EX DE,HL
CA02 29             301    ADD HL,HL
CA03 19             302    ADD HL,DE    ;Y*512*2+X*2
CA04 ED 5B A4 E0    303    LD DE,(#BADR)
CA08 19             304    ADD HL,DE
CA09 22 29 C8       305    LD (ADR2),HL
CA0C C9             306    RET
CA0D                307
CA0D                308   ;
CA0D                309   SETHLE ;OUT HL,E
CA0D 3A 2C C8       310     LD A,(RGBP)
CA10 2A 27 C8       311     LD HL,(ADR)
CA13 84             312     ADD A,H
CA14 67             313     LD H,A
CA15 3A 2B C8       314     LD A,(BIT)
CA18 5F             315     LD E,A
CA19 C9             316     RET
CA1A                317
```

THE SENTINEL

WZDとWLKの使い心地はいかがでしょうか。かえってプログラム作成の手間が増えたような気分？

それは単純なひとつのファイルだけから成るプログラムを作成しているからではないでしょうか。リロケータブルオブジェクトがおいしいのは，複数のファイルから成るプログラムを作成するときなのです。

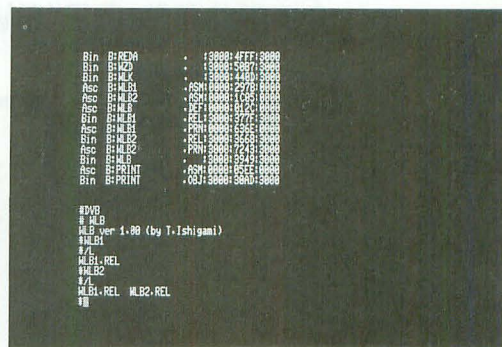
プログラムの作り方は大きく2通りに分けることができます。ひとつはトップダウンに作成する方法。まずメインルーチンを作り，次に必要となったサブルーチンを揃え，さらにその中で使ったルーチンを補っていくという作り方です。プログラムのモジュール化が図りやすいやり方です。

もうひとつはボトムアップに作成する方法。こちらはまずサブルーチンを用意し，次にそのサブルーチンを使ってさらに上位のサブルーチンを作り，最後にそれらをまとめて統轄するメインルーチンを作ります。サブルーチンごとにデバッグできるというメリットがあり，プログラムはデバッグの済んだサブルーチンの組み合わせで作れるわけですから，バグの出にくいプログラミング方法だといえるでしょう。

第99部
ライブラリアンWLB

WZD, WLKを使うと，特に後者のようなプログラム開発が容易になります。サブルーチンごとにひとつのファイルを作り，それとは別にチェック用のプログラムを用意します。そしてアセンブル・リンクして動作チェックし，まずいところがあればサブルーチンを修正・再アセンブルしてリンク。別の条件を試したければチェックルーチンのほうを書き直して再アセンブル・リンク。いずれももう一方はアセンブルする必要ありません。OKとなればチェックルーチンを削除。これで動くようになったサブルーチンがアセンブル終了の状態で残ることになります。

こうしてサブルーチンをRELファイルの状態で溜めながらプログラムを完成させていくわけです。アセンブル時間を短縮することができ，しかもデバッグを進めながら効率よくプログラムでき，さらにはサブルーチンも溜めることができます。

ただS-OSではどんなに小さなファイルでも最低4Kバイトのサイズとなってしまいますので，サブルーチンの数が増えてくると大変です。そこでWLBの登場です。これは複数のRELファイルを1個のLIBファイルにまとめてしまうものです。これでサブルーチンの管理がやりやすくなることでしょう。

●S-OSの系譜（14）

ダンプリストはマシン語のコードにチェックサムを付加した形で掲載されます。チェックサムはコードを足し合わせた答えの下2桁を取り出したもので，コードの入力間違いを発見しやすいように考えられたものです。1986年12月号までは，ダンプリストの横8バイトの合計を出した横サムと，ダンプリストを縦に16バイト合計した縦サムの2つのチェックサムを付けて掲載されていました。ダンプリストの右下に置かれたチェックサムはトータルサムと呼ばれ，1ブロックのデータ全部を足し合わせた答え，すなわち縦サムの合計の下2桁を取り出したもの(当然横サムの合計でもある)でした。ダンプリストを入力したらまずトータルサムを確認し，違っていれば縦サムあるいは横サムを見て間違っている場所を発見するという方法がダンプリスト入力の基本的な方法だったのです。

トータルサムは単に縦サム（横サムを）足し合わせたものですから，ダンプリストの2行を逆に入力してもミスを指摘してはくれません。トータルサムが合っていても，縦サムか横サムが違っているということはかなり頻繁に起きていました。さらに，当時のダンプリストでは，4つのデータを間違えるとチェックサムすべてが合ってしまう場合があるという問題点も抱えていました。そもそもトータルサムは単に縦サムの合計を表したものにすぎないのですから，縦サム・横サムの付加されたダンプリストには無用の長物ともいえるものです。そこでこのトータルサムをもっと有効なデータに置き換えられないかという議論がなされました。

この結果登場したのがCRCです。これは通信分野で使われているデータチェック方法のひとつで，特殊な割り算を行った余りを表しています。これをトータルサムの代わりに使ったダンプリストがOh!MZの標準ダンプリスト形式になり，入力ツールが1987年1月号のSENTINELで掲載されました。MACINTO-CはOh!Xに受け継がれ，今日も現役で使われています。

全機種共通S-OS“SWORD”要

ライブラリアンWLB

ライブラリアンを加えて，S-OSのリロケータブルオブジェクトによる開発システムも一段落。今後はこれらのシステムを使ったコンパイラなどを発表することになるでしょう。さあ，C言語が動くのはいつの日か？

Ishigami Tatsuya
石上 達也

いま合唱のサークルでヘンデルの「メサイア」という曲に取り組んでいるんですが，その中にこんな曲がありました。

Ev’ry valley shall be exalted (No.3)
Every valley shall be exalted,and every mountain and hill made low;
the crooked straight,and rough place plam. [Isaiah 40-4]
（ もろもろの谷は高くせられ（第3曲）
全ての谷は高くされ，
全ての山と丘とは，低くされる。
高低のある地は平らになり
険しいところは平地となる
「イザヤ伝40章-4」）

うーん，なにか，最近の私の周りの出来事に似ていると思えるのは気のせいだろうか？ 選抜選手のみなさん，ポピュラス大会残念でした。

ライブラリアンてなんじゃろな？

リロケータブルアセンブラを使ってプログラミングすることの利点のひとつに，

「すでに一度作成したサブルーチン（モジュール）の使い回しがきく」

ということがあります。

しかし，手持ちのモジュールが多くなると今度は，リロケータブルファイルでも管理に困り始めます。

理屈のうえでは，WZDとWLKがあれば，リロケータブルアセンブラを用いたプログラム開発は可能なのですが，リンクするファイルが多くなってくると，非常に面倒なことになってきます。ましてや，それが毎回毎回同じようなファイルだったりしたら，プログラムを作っているのか，それともキー入力の反復練習をしているのかわからなくなってしまいかねません（“SWORD”では，一応バッチ処理ができるんですけどね……。誰かMAKEを作ってくれないかなぁ？）。

そのほかにも，

●どんなに小さなモジュールでもディスクに収めれば必ず，1クラスタ（4Kバイト相当）を占有する。ただでさえ，あまり容量の大きくない2Dのシステムに，このことはたいへん不利である。

●1つひとつのモジュールに対していちいちファイルのオープン&クローズを行っていたのでは，速度的に不経済である。

●末葉的なモジュールは人間が手動で管理するよりも，なんらかの抽象的なシンボルでまとめて管理したほうが，OOPS (Object Oriented Programming Systemうーん？)していて，カッコイイじゃないかい？

などの理由により，モジュールを（半ば）自動的に管理してくれるようなシステムがほしくなってくるわけで，そこでライブラリアンの御登場となるわけです。

まあ以上の説明で，実感を持てるのは，リロケータブルアセンブラを使い込んでいて，モジュールの管理に心底困ったことのある方でしょう。もっと，ぐっと簡単に説明してしまうと，

「Oh!Xが記事ごとに（モジュールごとに）書店に並んでいて，読者（リンカ）は，読みたい記事のみをそこから選んで買っていき，各自がファイルするなり，床に積み上げておくなりして管理するよりも，Oh!Xとして1冊にまとめて売っていたほうが，（必要のない記事（んっ？）の分だけ量は増えるが，）管理が楽である。」

ということです。そして，編集部の代わりに編集を行うのがライブラリアンなのです。ここで，もう一度前述の説明を読んでみてください。これで，だいたいのイメージはできたかと思います。

WLBの使い方

起動を終えてプロンプト'*'が表示されているのを確認してください。次に，ひとつに束ねてしまいたいファイル（リロケータブルファイルであっても，ライブラリファイルであっても構いません）の名前をポコポコと入力します。このとき，ファイル名は1つひとつ入力してリターンキーを押してもよいし，いくつかのファイル名を，','で区切りつつ，まとめて入力してリターンキーを押しても構いません。

ファイル名の拡張子が省略された場合，まず'.REL'でファイルを捜し，そのファイルが存在しない場合には'.LIB'を捜します。

そして，束ねてしまいたいファイルの名前をすべて入力し終わったら，それらを出力します。出力したいファイルの名前をタイプし，その後ろに'/E'をつけてリターンキーを押します。

たとえば，AAA.RELとBBB.RELをくっつけてCCC.LIBというライブラリファイルを作ろうとします。このときは以下のようにします。

```
#LWLB
#J3000   (WLBを起動する)
*AAA     (AAAを読み込む)
*BBB     (BBBを読み込む)
*CCC/E   (CCCを書き出す)
#シフト+ブレイクキーを押す
```

これで，ディスク上にはCCC.LIBというファイルができあがっているはずです。

注意点

WLBは，作業中にカレントディスクに$$$$$$$$.TMPというテンポラリファイルを作成し，作業の終了と共に消去します。したがって，$$$$$$$$.TMPという名のファイルをカレントディスクに持っていると自動的に消去されてしまいますので，注意してください。同様の理由により，作業中にシフト+ブレイクで強制的にWLBを抜け出した場合，カレントディスクに$$$$$$$$.TMPというファイルが残ってしまう場合がありますので，そのときは，S-OSのKコマンドで消去してください。

また，RAMディスクをお持ちの方は，カレントディスクにRAMディスクを指定しておくと作業が早まります。

プログラムについて

このプログラムは以下の4本のファイルから成っています。

●SOS.DEF（S-OSシステムのシステムルーチンの定義ファイル）

●WLB.DEF（WLBで用いる定数の定義ファイル）

●WLB1.ASM（主に人間からの指示を解読する）

●WLB2.ASM（ファイル操作など）

ここに，掲載されているソースプログラムからアセンブルするときには，

```
#LWZD
#J3000
*=WLB1
*=WLB2
*［シフト＋ブレイク］でWLBを抜け
```

出す。

で，アセンブルをして，さらに，

```
#LWLK
#J3000
*/P:3000,/D:4500
*WLB1,WLB2,WLB/N:P
```

でオブジェクトプログラムが得られます。

プログラムの中身については，ほとんどリンカと同じです。ただ，読み込んだリロケータブルファイルを，実際に，オブジェクトとしてつなげていくか，それとも，リロケータブルファイルのまま出力するかが違うだけです。WLKを解析しようという方は，WLBを先に解析するとよいでしょう。

WLBはすべての入力が終わった時点で束ねる処理を行います。入力ファイルの名前が入力されると，ディスク上に存在するか否か，とそのファイルサイズを記憶しておきます。入力ファイルがライブラリファイルの場合には含まれるモジュールすべても記憶しておきます（表1参照）。

そして出力ファイルが指定されると，いままでの記憶した情報をもとに，実際の束ねる作業を行います。これは，S-OS上で使用できるメモリがかなり限られているためです。

この方法にしたため以下のような欠点が出てきてしまいました。

1）　同じファイルを2度もオープンするので処理速度が遅い

2）　束ねる作業中にディスクを交換できない。つまり，ディスクを2ドライブしか持っていないユーザーは，リロケータブルファイルを2枚以内に集めておかなくてはならない。

まぁ，1）の欠点はしょうがないにしても，2）の欠点は，困ったものです。とりあえず，WLBはライブラリファイルも入力ファイルとして受け付けますので，ディスクごとにライブラリファイルを作って，最後にまとめるなどの方法で逃げてください（この欠点はWLKにも共通です。リンカにそんなに多くのファイルを読み込ませることはないので，意識する必要はないと思いますが）。

＊　＊　＊

先月号ですっかり書くのを忘れてしまいましたが，WZD＆WLKを使って作成したプログラムを，1987年11月号で発表されたファイルアロケータ＆ローダのフォーマットに変換する方法を紹介します。

わかってしまえばなんのことはない，オブジェクトプログラムを2種類作ってしまえばよいのです。

具体的にいうと，いまPROG.ASMというソースファイルをアセンブルするとします。

```
# WZD =PROG
```

で，リロケータブルファイルを作成して，

```
# WLK /P:0000, PROG,PROG1/N
```

として，アドレス0000Hから始まるPROG1.OBJを用意します。次に，

```
# WLK /P:1234, PROG,PROG2/N
```

として，アドレス1234Hから始まるPROG2.OBJを用意します。あとはもう，説明の必要がないでしょう。

老婆心ながら，アロケータは，

```
L1: LD A,L1*2
```

などというプログラムに対しては，対応していませんので注意してください（アドレス情報を掛けたり，割ったりすることはないと思いますが，念のため）。

表1　WLBのコマンドの格納のされ方

01H，［ファイルサイズ（2バイト）］，"ファイル名"，00H
　RELファイルを読み込む。
02H，"ファイル名"，00H
03H，［ファイルサイズ（2バイト）］，［シークアドレス（2バイト）］，"モジュール名"，00H
　モジュールを読み込む。
04H，［ファイルサイズ（2バイト）］，"ファイル名"，00H
　消去された01コマンド
05H，［ファイルサイズ（2バイト）］，［シークアドレス（2バイト）］，"モジュール名"，00H
　消去された03コマンド
FFH
　コマンドの終わり。

表2　リファレンスマニュアル

起動方法

"SWORD"の拡張をしていない人はコマンドラインから

```
# LWLB
# J3000
```

と，拡張をしてある人は，

```
# WLB
```

で起動します。すると，このあとプロンプト'*'を表示してパラメータの入力待ちになります。なお，それぞれ

```
# J3000
# WLB
```

の後ろに，パラメータを書くこともできます（ここらへんは，先月号のWZDと同じです）。

パラメータ

まず，基本形は，

```
＊［ファイル名］
```

です。この動作の繰り返しによって，組み込みたいモジュールやライブラリファイルなどを指定していきます。そして，組み込みたいモジュールをすべて指定し終わったら，

```
＊［ファイル名］/E
```

とすると，いままで指定してきたモジュールを組み込んだライブラリファイルが［ファイル名］で表されたファイル名で作成されます。

スイッチ

＊/C
　WLBの初期化を行う。
＊/U
　未定義ラベルの一覧表を画面に表示する。
＊/L
　定義済みのラベルの一覧表を画面に表示する。
＊［ファイル名］/E
　LIBファイルとして，［ファイル名］を出力する。省略時の拡張子は'.LIB'である。
＊［ファイル名］/R
　LIBファイルとして，［ファイル名］を出力する。
＊［シフト＋ブレイク］
　WLBの一切の作業を中断して，S-OSのコマンドレベルに戻る。

エラーメッセージ

?Err
　WLBに対するコマンドがおかしい。
File Access Error On ［ファイル名］
　［ファイル名］のファイルにアクセス中にエラーが発生した。
Illegal File
　与えられたファイルはリロケータブルファイルでも，ライブラリファイルでもない
File Not Found
　該当するファイルが見当たらない。
その他は，"SWORD"のエラーメッセージに準じています。

リスト1 WLB

```
3000 ED 7B 6A 1F 21 BA 35 CD : CE
3008 B1 35 ED 5B 76 1F 13 13 : E9
3010 06 01 1A 13 A7 28 07 FE : 08
3018 20 20 F7 04 18 F4 C5 DD : E9
3020 21 00 50 DD 36 00 FF F1 : 74
3028 FE 01 20 05 CD 8A 34 18 : C7
3030 11 FD 2A 76 1F FD 23 FD : EA
3038 23 FD 7E 00 FD 23 FE 20 : DC
3040 20 F7 FD 7E 00 A7 CC 8A : 8F
3048 34 11 00 45 FD E5 E1 CD : 1A
3050 13 35 E5 FD E1 21 E5 30 : 41
3058 CD AA 34 21 E8 30 C4 AA : 52
3060 34 20 0B DD 21 00 50 DD : 8A
3068 36 00 FF C3 C8 30 21 F1 : 02
3070 30 CD AA 34 21 F4 30 C4 : E4
3078 AA 34 20 06 CD E1 31 C3 : A6
---------------------------------
SUM: 8F D4 6A A4 12 81 90 67 7E0E

3080 C8 30 21 EB 30 CD AA 34 : DF
3088 21 EE 30 C4 AA 34 20 0A : 0B
3090 DD E5 CD 94 32 DD E1 C3 : D6
3098 C8 30 21 F7 30 CD AA 34 : EB
30A0 21 FA 30 C4 AA 34 20 06 : 13
30A8 CD 94 32 C3 FA 1F 21 FD : 8D
30B0 30 CD AA 34 21 00 31 C4 : F1
30B8 AA 34 20 06 CD 2A 32 C3 : F0
30C0 C8 30 CD 11 31 DC 33 20 : 36
30C8 FD 7E 00 A7 28 14 FE 2C : 88
30D0 FD 23 28 0E 21 0B 31 CD : 80
30D8 B1 35 FD 2A 76 1F FD 36 : D5
30E0 00 00 C3 42 30 2F 43 00 : A7
30E8 2F 63 00 2F 45 00 2F 65 : 9A
30F0 00 2F 4C 00 2F 6C 00 2F : 45
30F8 52 00 2F 72 00 2F 4B 00 : 6D
---------------------------------
SUM: 4A 5A 9B CE 62 0C 15 A2 CC27

3100 2F 6B 00 52 45 4C 00 4C : C9
3108 49 42 00 3F 45 72 72 0D : 00
3110 00 21 00 45 CD D8 34 11 : 50
3118 00 45 21 03 31 DC E2 34 : 8C
3120 3E 01 11 00 45 CD A3 1F : 24
3128 D8 CD 09 20 38 21 DD 36 : 3A
3130 00 01 DD 23 2A 72 1F DD : 99
3138 75 00 DD 23 DD 74 00 DD : A3
3140 23 DD E5 D1 21 00 45 CD : E9
3148 FC 34 E5 DD E1 B7 C9 FE : 51
3150 08 37 C0 11 00 45 21 07 : 7D
3158 31 CD E2 34 3E 01 11 00 : 64
3160 45 CD EC 35 D8 DD 36 00 : 1E
3168 02 DD 23 DD E5 D1 21 00 : B6
3170 45 CD FC 34 E5 DD E1 21 : 06
3178 00 00 22 2A 45 21 2D 45 : 24
---------------------------------
SUM: E7 6E 8E A2 33 EF CC E5 C0DA

3180 22 2C 45 CD 9E 37 FE FF : 32
3188 CA CD 31 FE E0 C2 25 35 : C2
3190 DD 36 00 03 CD 9E 37 DD : 95
3198 77 03 CD 9E 37 DD 77 04 : 74
31A0 67 DD 6E 03 ED 5B 2A 45 : 6C
31A8 22 2A 45 B7 ED 52 EB 2A : 9C
31B0 2C 45 23 73 23 72 DD 22 : 9B
31B8 2C 45 11 05 00 DD 19 CD : 4A
31C0 9E 37 DD 77 00 DD 23 A7 : D0
31C8 20 F5 C3 83 31 2A 49 45 : 44
31D0 ED 5B 2A 45 B7 ED 52 EB : 98
31D8 2A 2C 45 23 73 23 72 B7 : 7D
31E0 C9 DD 36 00 FF 21 00 50 : 4C
31E8 7E 23 FE FF CA EB 1F FE : 70
31F0 01 CC 16 32 FE 02 CC 0E : EF
31F8 32 FE 03 CC 14 32 FE 04 : 47
---------------------------------
SUM: 70 40 86 FD B5 C7 F5 61 7BB2

3200 CC 0C 32 FE 05 CC 0A 32 : 15
3208 18 DE 23 23 23 23 7E 23 : 23
3210 A7 20 FB C9 23 23 23 23 : 17
3218 CD B1 35 D5 ED 5B 7A 1F : 69
3220 1A D1 E6 07 C8 CD F1 1F : 7D
3228 18 F1 DD 36 00 FF 21 00 : 3C
3230 50 22 28 45 7E 23 FE FF : 7D
3238 28 2B FE 01 CC 7A 32 28 : F2
3240 18 FE 02 CC 0E 32 FE 03 : 25
3248 CC 78 32 28 12 FE 04 CC : 7E
3250 0C 32 FE 05 CC 0A 32 18 : 61
3258 D8 2A 28 45 36 04 C9 2A : 9C
3260 28 45 36 05 C9 21 6C 32 : 30
3268 CD B1 35 C9 4E 6F 74 20 : CD
3270 46 6F 75 6E 64 20 0D 00 : 29
3278 23 23 23 23 11 00 45 1A : FC
---------------------------------
SUM: 28 24 CB DF F8 C4 96 5A 752F

3280 A7 28 07 BE 20 07 13 23 : F1
3288 18 F5 7E A7 C9 CD 0E 32 : 08
3290 3E 01 A7 C9 DD 36 00 FF : C1
3298 11 14 45 21 00 45 CD FC : 99
32A0 34 21 14 45 CD D8 34 11 : 98
32A8 14 45 21 07 31 DC E2 34 : A4
32B0 3E 01 11 14 45 CD A3 1F : 38
32B8 D8 3A 5D 1F 32 D8 35 3E : 0B
32C0 01 11 D8 35 CD D6 36 D8 : D0
32C8 CD F5 32 22 30 45 CD 5F : B7
32D0 33 CD F5 33 CD 2C 37 D8 : 30
32D8 11 14 45 CD A3 1F D8 CD : 9E
32E0 7F 36 D8 CC 15 20 D8 11 : 77
32E8 D8 35 CD A3 1F D8 11 14 : 99
32F0 45 CD 12 20 C9 DD 21 00 : 0B
32F8 50 21 01 00 DD 7E 00 DD : AA
---------------------------------
SUM: 6A 13 10 B4 82 61 F8 D0 7406

3300 23 FE FF C8 FE 01 20 0C : 13
3308 DD 23 DD 23 23 23 23 CD : 36
3310 43 33 18 E8 FE 02 20 05 : 9B
3318 CD A1 34 18 DF FE 03 20 : BA
3320 10 DD 23 DD 23 DD 23 DD : ED
3328 23 23 23 23 CD 43 33 18 : E7
3330 CB FE 04 28 04 DD 23 DD : D6
3338 23 DD 23 DD 23 CD A1 34 : C5
3340 C3 FC 32 DD 7E 01 FE 3A : 85
3348 20 04 DD 23 DD 23 DD 7E : 7F
3350 00 DD 23 23 FE 2E CA A1 : BA
3358 34 CD 78 35 20 F0 C9 DD : 64
3360 21 00 50 2A 30 45 22 32 : 64
3368 45 DD 7E 00 DD 23 FE FF : 9D
3370 CA 84 34 FE 01 20 32 3E : 11
3378 E0 CD FD 37 3A 32 45 CD : 5F
---------------------------------
SUM: 58 A8 3E A7 D6 EA 85 76 26E9

3380 FD 37 3A 33 45 CD FD 37 : E7
3388 2A 32 45 DD 5E 00 DD 23 : DC
3390 DD 56 00 DD 23 19 22 32 : A0
3398 45 DD 7E 01 FE 3A 20 04 : FD
33A0 DD 23 DD 23 CD BC 34 18 : D5
33A8 C0 FE 02 20 06 CD A1 34 : 88
33B0 C3 69 33 FE 03 20 2C 3E : EA
33B8 E0 CD FD 37 3A 32 45 CD : 5F
33C0 FD 37 3A 33 45 CD FD 37 : E7
33C8 2A 32 45 DD 5E 00 DD 23 : DC
33D0 DD 56 00 DD 23 19 22 32 : A0
33D8 45 DD 23 DD 23 CD BC 34 : 02
33E0 C3 69 33 FE 04 28 04 DD : 6A
33E8 23 DD 23 DD 23 DD 23 CD : F0
33F0 A1 34 C3 69 33 DD 21 00 : 32
33F8 50 DD 7E 00 DD 23 FE FF : A8
---------------------------------
SUM: A9 E6 45 74 F4 B3 60 50 D378

3400 C8 FE 01 20 32 DD 4E 00 : 44
3408 DD 23 DD 46 00 DD 23 C5 : E8
3410 DD E5 E1 11 00 45 CD 13 : D9
3418 35 23 E5 DD E1 3E 01 11 : 4B
3420 00 45 CD EC 35 C1 D8 C5 : 91
3428 CD 9E 37 CD FD 37 C1 0B : 6F
3430 78 B1 20 F3 C3 F9 33 FE : 29
3438 02 20 19 DD E5 E1 11 00 : EF
3440 45 CD 13 35 23 E5 DD E1 : 20
3448 3E 01 11 00 45 CD EC 35 : 83
3450 D8 C3 F9 33 FE 03 20 1C : 04
3458 DD 6E 02 DD 66 03 CD B2 : 12
3460 37 DD 4E 00 DD 46 01 C5 : 4B
3468 CD 9E 37 CD FD 37 C1 0B : 6F
3470 78 B1 20 F3 FE 04 28 04 : 6A
3478 DD 23 DD 23 DD 23 DD 23 : 00
---------------------------------
SUM: 8F 2B 82 05 6E 6B 99 92 8C0A

3480 CD A1 34 C9 3E FF CD FD : 72
3488 37 C9 3E 2A CD F4 1F ED : 35
3490 5B 76 1F CD D3 1F 1A FE : C7
3498 1B CA FA 1F 13 D5 FD E1 : C4
34A0 C9 DD 7E 00 DD 23 A7 20 : EB
34A8 F8 C9 FD E5 D1 7E A7 28 : C1
34B0 07 1A BE C0 13 23 18 F5 : E2
34B8 D5 FD E1 C9 DD 7E 00 FE : D5
34C0 2E 28 0C CD FD 37 DD 7E : BE
34C8 00 DD 23 A7 C8 18 ED CD : 41
34D0 A1 34 3E 00 CD FD 37 C9 : DD
34D8 7E 23 A7 37 C8 FE 2E 20 : 93
34E0 F7 C9 06 0F 1A CD 78 35 : 69
34E8 38 0A 05 28 07 FE 2E 28 : CA
34F0 03 13 18 F0 3E 2E 12 13 : AF
34F8 CD FC 34 C9 06 14 AF 12 : A1
---------------------------------
SUM: 63 A5 10 E8 4E 80 FF BA D01F

3500 7E 23 CD 78 35 38 09 05 : 61
3508 28 06 12 13 AF 12 18 F0 : 1C
3510 EB 23 C9 06 14 AF 12 7E : 30
3518 CD 78 35 D8 05 C8 23 12 : 54
3520 13 AF 12 18 F2 21 4B 35 : 7F
3528 CD B1 35 21 00 45 CD B1 : 97
3530 35 CD EE 1F C9 46 69 6C : F3
3538 65 20 41 63 63 65 73 73 : D7
3540 20 45 72 72 6F 72 20 6F : B9
3548 6E 3A 00 49 6C 6C 65 67 : 95
3550 61 6C 20 66 69 6C 65 20 : AD
3558 0D 00 42 61 64 20 43 6F : E6
3560 6D 6D 61 6E 64 20 0D 00 : 3A
3568 46 69 6C 65 20 6E 6F 74 : F1
3570 20 66 6F 75 6E 64 0D 00 : 49
3578 FE 2F 28 06 FE 2C 28 02 : AF
---------------------------------
SUM: A5 67 8B F4 B3 5A 28 25 97D4

3580 A7 C0 37 C9 21 00 00 FD : 85
3588 7E 00 16 30 FE 30 D8 FF : C8
3590 3A 38 11 16 41 FE 41 D8 : F1
3598 FE 47 38 08 16 61 FE 61 : 5B
35A0 D8 FE 67 D0 29 29 29 29 : B1
35A8 92 16 00 5F 19 FD 23 18 : 58
35B0 D6 7E 23 A7 C8 CD F1 1F : C6
35B8 18 F7 57 4C 42 20 76 65 : EF
35C0 72 20 31 2E 30 30 20 28 : 99
35C8 62 79 20 54 2F 49 73 68 : A1
35D0 69 67 61 6D 69 29 0D 00 : 3D
35D8 41 3A 57 4C 42 20 20 20 : C0
35E0 20 20 20 20 20 20 20 2E : 0E
35E8 54 4D 50 00 CD A3 1F D8 : 58
35F0 CD 55 36 D8 2A 74 1F 11 : FE
35F8 A5 45 01 20 00 ED B0 3A : E2
---------------------------------
SUM: 19 09 27 8C E2 88 9B FA 012D

3600 5D 1F 32 D5 45 AF 32 36 : DF
3608 45 CD E5 38 D8 06 10 0E : 2B
3610 00 3A C3 45 11 C5 45 12 : 6F
3618 13 FE 7F 30 0F 2A 62 1F : 7A
3620 85 6F 30 01 24 7E 05 28 : F4
3628 28 0C 18 EB 0D F5 79 87 : 39
3630 87 87 87 4F F1 D6 80 81 : AC
3638 32 DB 45 AF 32 D6 45 01 : 4F
3640 37 00 11 37 45 21 A5 45 : CF
3648 ED B0 3E FF 32 C4 37 B7 : BE
3650 C9 3E 07 37 C9 3A 5D 1F : C4
3658 CD 74 36 D8 CD 7F 36 D8 : A9
3660 3E 08 37 C0 E5 ED 5B 74 : DE
3668 1F 01 20 00 ED B0 E1 7E : 3C
3670 CD C9 36 C9 FE 41 38 04 : 10
3678 FE 45 3F D0 3E 03 C9 0E : 6A
---------------------------------
SUM: FD 7A C5 0A AC 42 D8 9D 36EC

3680 10 ED 5B 60 1F ED 53 D7 : EE
3688 45 2A 64 1F 3E 01 CD 00 : FE
3690 20 D8 06 08 22 D9 45 7E : C4
3698 FE FF 28 1A B7 28 0B D5 : FE
36A0 ED 5B 74 1F CD BA 36 D1 : 69
36A8 28 0D D5 11 20 00 19 D1 : 25
36B0 10 E2 13 0D 20 CF 3E AF : EE
36B8 B7 C9 C5 E5 06 10 13 23 : 76
36C0 1A BE 20 02 10 F8 E1 C1 : A4
36C8 C9 E5 E6 87 21 1F 29 BE : 42
36D0 E1 C8 3E 06 37 C9 CD A3 : 5D
36D8 1F D8 CD AF 1F D8 01 20 : 8B
36E0 00 11 A5 45 2A 74 1F ED : A5
36E8 B0 3A 5D 1F 32 D5 45 2A : DC
36F0 E1 27 22 D9 45 2A DF 27 : 78
36F8 22 D7 45 CD 14 39 D8 32 : 62
---------------------------------
SUM: E5 8D 88 0B 85 EC 03 50 570C

3700 C3 45 32 C5 45 3E 80 32 : 34
3708 C6 45 AF 32 DB 45 32 36 : 74
3710 45 32 D6 45 21 00 00 22 : D5
3718 B7 45 01 37 00 11 6E 45 : F8
3720 21 A5 45 ED B0 3E 00 32 : 18
3728 19 38 B7 C9 21 6E 45 11 : B6
3730 A5 45 01 37 00 ED B0 3A : F9
3738 D5 45 32 5D 1F 2A B7 45 : EE
3740 2C 2D C4 2C 38 3A DB 45 : DB
3748 2A B7 45 2C 2D 20 01 3C : DC
3750 67 22 B7 45 3E 01 ED 5B : 0C
3758 D7 45 2A 64 1F CD 00 20 : B6
3760 D8 21 A5 45 ED 5B D9 45 : 49
3768 01 20 00 ED B0 3E 01 ED : EA
3770 5B D7 45 2A 64 1F CD 03 : F4
3778 20 D8 CD E5 38 D8 06 10 : D0
---------------------------------
SUM: 21 A3 88 FF 2C 0F 42 D2 D36A

3780 21 C5 45 7E FE 7F 30 12 : 68
3788 23 4E E5 2A 62 1F 16 00 : 17
3790 5F 19 71 E1 05 CA 51 36 : 20
3798 18 E9 CD 03 39 C9 21 C4 : B8
37A0 37 34 20 08 CD C6 37 D8 : 35
37A8 21 68 45 34 2A C4 37 7E : A5
37B0 B7 C9 7D 3D 32 C4 37 7C : E3
37B8 32 68 45 CD C6 37 21 68 : 32
37C0 45 34 B7 C9 00 B0 01 37 : E1
37C8 00 11 A5 45 21 37 45 ED : 85
37D0 B0 3A D5 45 32 5D 1F 3A : EC
37D8 D6 45 47 3A DB 45 B8 DA : 4E
37E0 51 36 78 CD C9 38 EB 3E : F6
37E8 01 21 00 B0 CD 00 20 D8 : 97
37F0 01 37 00 11 37 45 21 A5 : 8B
37F8 45 ED B0 B7 C9 2A 80 45 : 51
---------------------------------
SUM: 5F 21 2F A4 51 E6 47 7E 928F

3800 23 22 80 45 2A 19 38 77 : FC
3808 21 19 38 34 37 3F C0 CD : A9
3810 1B 38 D8 21 9F 45 34 B7 : 1B
3818 C9 00 B1 01 37 00 11 A5 : 68
3820 45 21 6E 45 ED B0 3A D5 : C5
3828 45 32 5D 1F 3A D6 45 47 : 8F
3830 3A DB 45 B8 30 76 CD E5 : 6A
3838 38 D8 3A D6 45 E6 F0 47 : 82
3840 3A DB 45 E6 F0 B8 30 47 : 5F
3848 CB 3F CB 3F CB 3F CB 3F : 28
3850 21 C6 45 16 00 5F 19 E5 : 9F
3858 3A DB 45 E6 F0 CD C9 38 : FE
3860 CD 36 39 2A 62 1F 16 00 : FD
3868 5F 19 36 8F CD 14 39 77 : CE
3870 E1 D8 77 23 36 80 2A 62 : 95
3878 1F 16 00 5F 19 36 80 3A : 9D
---------------------------------
SUM: B0 71 0B E9 FC 8B 4F 9E 7B54

3880 DB 45 F6 F0 C6 10 32 DB : D9
3888 45 CD 03 39 D8 18 9D 3A : 15
3890 D6 45 32 DB 45 CB 3F CB : 42
3898 3F CB 3F CB 3F 21 C6 45 : 7F
38A0 16 00 5F 19 3A D6 45 E6 : C9
38A8 0F C6 80 77 3A D6 45 CD : EE
38B0 C9 38 EB 3E 01 21 00 B1 : FD
```

▶つい先日フラクタルの記事を読もうと思い，昔のOh!Xを開いた。ら，ぺりぺり！
平木　俊行 (19) 神奈川県

```
38B8 CD 03 20 D8 01 37 00 11 : 11
38C0 6E 45 21 A5 45 ED B0 B7 : 12
38C8 C9 F5 F5 CB 3F CB 3F CB : 92
38D0 3F CB 3F 21 C5 45 16 00 : 8A
38D8 5F 19 7E CD 2E 39 F1 E6 : 01
38E0 0F 85 6F F1 C9 D5 E5 3A : B1
38E8 36 45 57 3A 5D 1F BA 28 : 6A
38F0 0F 32 36 45 3E 01 ED 5B : 43
```

```
38F8 5E 1F 2A 62 1F CD 00 20 : 15
----------------------------------
SUM: 77 5C 3D A5 92 10 E0 DF AD82

3900 E1 D1 C9 D5 E5 3E 01 ED : 61
3908 5B 5E 1F 2A 62 1F CD 03 : 53
3910 20 E1 D1 C9 C5 E5 06 80 : CB
3918 2A 62 1F 7E B7 28 08 23 : 33
```

```
3920 10 F9 3E 09 37 18 04 3E : E1
3928 80 90 A7 E1 C1 C9 26 00 : 48
3930 6F 29 29 29 29 C9 E5 CB : 8C
3938 3C CB 1D CB 3C CB 1D CB : DE
3940 3C CB 1D CB 3C CB 1D 7D : 90
3948 E1 C9                   : AA
----------------------------------
SUM: DE 83 20 EF 5C AA 25 E4 7D40
```

リスト2 WLB1.ASM

```
  1 0000'                   ;===================================
  2 0000'                   ;
  3 0000'                   ; Librarian For W-ZEDA
  4 0000'                   ;       Programed by T.Ishigami
  5 0000'                   ;               '90 Jul 10th
  6 0000'                   ;
  7 0000'                   ;===================================
  8 0000'
  9                                 ORG     3000H
 10 3000'
 11 3000'                 C         INCLUDE SOS.DEF
 12 1FFA                  C _HOT    EQU     1FFAH
 13 1FF4                  C _PRINT  EQU     1FF4H
 14 1FF1                  C _PRNTS  EQU     1FF1H
 15 1FEE                  C _LTNL   EQU     1FEEH
 16 1FEB                  C _NL     EQU     1FEBH
 17 1FE5                  C _MSX    EQU     1FE5H
 18 1FDF                  C _TAB    EQU     1FDFH
 19 1FD3                  C _GETL   EQU     1FD3H
 20 1FC7                  C _PAUSE  EQU     1FC7H
 21 1FC1                  C _PRTHX  EQU     1FC1H
 22 1FBE                  C _PRTHL  EQU     1FBEH
 23 1FB8                  C _HEX    EQU     1FB8H
 24 1FA3                  C _FILE   EQU     1FA3H
 25 1FAF                  C _WOPEN  EQU     1FAFH
 26 1F9A                  C _POKE   EQU     1F9AH
 27 1F94                  C _PEEK   EQU     1F94H
 28 2009                  C _ROPEN  EQU     2009H
 29 2015                  C _KILL   EQU     2015H
 30 2012                  C _NAME   EQU     2012H
 31 2033                  C _ERROR  EQU     2033H
 32 3000'                 C
 33 2000                  C _DRDSB  EQU     2000H
 34 2003                  C _DWTSB  EQU     2003H
 35 3000'                 C
 36 1F7A                  C _PRCNT  EQU     1F7AH
 37 1F76                  C _KBFAD  EQU     1F76H
 38 1F74                  C _IBFAD  EQU     1F74H
 39 1F72                  C _SIZE   EQU     1F72H
 40 1F6A                  C _MEMAX  EQU     1F6AH
 41 1F64                  C _DTBUF  EQU     1F64H
 42 1F62                  C _FATBF  EQU     1F62H
 43 1F60                  C _DIRPS  EQU     1F60H
 44 1F5E                  C _FATPOS EQU     1F5EH
 45 1F5D                  C _DSK    EQU     1F5DH
 46 3000'                 C
 47 3000'                 C         INCLUDE WLB.DEF
 48 3000'                 C ;=============================
 49 3000'                 C ; WLB.DEF Header File For WLB
 50 3000'                 C ;     # WZD
 51 3000'                 C ;     #=WLB1
 52 3000'                 C ;     *=WLB2
 53 3000'                 C ;     = WLK /P:3000,/D:4500
 54 3000'                 C ;     * WLB1,WLB2,WLB/N:P
 55 3000'                 C ;     #
 56 3000'                 C ;     CSEG      3000H
 57 3000'                 C ;     DSEG      4500H
 58 3000'                 C ;
 59 3000'                 C ;=============================
 60 3000'                 C
 61 B000                  C RDBUF   EQU     0B000H  ;- B0FFH
 62 B100                  C WRBUF   EQU     0B100H  ;- B1FFH
 63 3000'                 C
 64 5000                  C cmdbuf  EQU     5000H
 65 3000'                 C
 66 3000'                   ;
 67 3000'                   ; Constnts
 68 3000'                   ;
 69 0001                    REL     EQU     1
 70 0002                    LIB     EQU     2
 71 0003                    MOD     EQU     3
 72 0004                    UREL    EQU     4       ;Deleted REL Command
 73 0005                    UMOD    EQU     5       ;Deleted MOD Command
 74 3000'
 75 000D                    CR      EQU     0DH
 76 001B                    BRK     EQU     1BH
 77 00FF                    EOF     EQU     0FFH
 78 3000'
 79 000F                    NAME    EQU     15
 80 0003                    EXT     EQU     3
 81 0014                    LENFL   EQU     NAME+1+EXT+1
 82 3000'
 83 3000'
 84 3000'                   ;
 85 3000'                   ;   Start
 86 3000'                   ;
 87 3000'                   ;
 88 3000' ED 7B 6A 1F               LD      SP,(_MEMAX)
 89 3004'
 90 3004' 21 BA 35                  LD      HL,TITLE
 91 3007' CD B1 35                  CALL    _puts                   ;Print Title
 92 300A'
 93 300A' ED 5B 76 1F               LD      DE,(_KBFAD)
 94 300E' 13                        INC     DE
 95 300F' 13                        INC     DE                      ;Skip '# ' or'#J'
 96 3010' 06 01                     LD      B,1                     ;B = argc
 97 3012' 1A                CC1:    LD      A,(DE)
 98 3013' 13                        INC     DE
 99 3014' A7                        AND     A
100 3015' 28 07                     JR      Z,CC3
101 3017' FE 20                     CP      ' '
102 3019' 20 F7                     JR      NZ,CC1
103 301B' 04                        INC     B
104 301C' 18 F4                     JR      CC1                     ;if (DE) = ' ' then argc++
105 301E'
106 301E' C5                CC3:    PUSH    BC
107 301F'
108 301F' DD 21 00 50               LD      IX,cmdbuf               ;IX means Command pointer
109 3023' DD 36 00 FF               LD      (IX),EOF
110 3027' F1                        POP     AF                      ;A = argc
111 3028' FE 01                     CP      1
112 302A' 20 05                     JR      NZ,CC6
113 302C'
114 302C' CD 8A 34                  CALL    GETL                    ;Get Command Line
115 302F' 18 11                     JR      CC8
116 3031'
117 3031' FD 2A 76 1F       CC6:    LD      IY,(_KBFAD)
118 3035' FD 23                     INC     IY
119 3037' FD 23                     INC     IY                      ;Skip '# ' or'#J'
120 3039' FD 7E 00          CC4:    LD      A,(IY)
121 303C' FD 23                     INC     IY
122 303E' FE 20                     CP      ' '
123 3040' 20 F7                     JR      NZ,CC4
124 3042'
125 3042' FD 7E 00          CC8:    LD      A,(IY)
126 3045' A7                        AND     A
127 3046' CC 8A 34                  CALL    Z,GETL
128 3049'
129 3049' 11 ** **                  LD      DE,inname
130 304C' FD E5                     PUSH    IY
131 304E' E1                        POP     HL              ;LD HL,IY
132 304F' CD 13 35                  CALL    fcopy2
133 3052' E5                        PUSH    HL
134 3053' FD E1                     POP     IY              ;LD IY,HL
135 3055'
136 3055' 21 E5 30                  LD      HL,swC
137 3058' CD AA 34                  CALL    amatch
138 305B' 21 E8 30                  LD      HL,swC+3
139 305E' C4 AA 34                  CALL    NZ,amatch
140 3061' 20 0B                     JR      NZ,CC10         ;'/C' Command
141 3063'
142 3063' DD 21 00 50               LD      IX,cmdbuf
143 3067' DD 36 00 FF               LD      (IX),EOF
144 306B' C3 C8 30                  JP      CC12
145 306E'
146 306E' 21 F1 30          CC10:   LD      HL,swL
147 3071' CD AA 34                  CALL    amatch
148 3074' 21 F4 30                  LD      HL,swL+3
149 3077' C4 AA 34                  CALL    NZ,amatch
150 307A' 20 06                     JR      NZ,CC17
151 307C'
152 307C' CD E1 31                  CALL    prtmod
153 307F' C3 C8 30                  JP      CC12
154 3082'
155 3082' 21 EB 30          CC17:   LD      HL,swE
156 3085' CD AA 34                  CALL    amatch
157 3088' 21 EE 30                  LD      HL,swE+3
158 308B' C4 AA 34                  CALL    NZ,amatch
159 308E' 20 0A                     JR      NZ,CC21         ;'/E' Command
160 3090'
161 3090' DD E5                     PUSH    IX
162 3092' CD 94 32                  CALL    putout
163 3095' DD E1                     POP     IX
164 3097' C3 C8 30                  JP      CC12
165 309A'
166 309A' 21 F7 30          CC21:   LD      HL,swR
167 309D' CD AA 34                  CALL    amatch
168 30A0' 21 FA 30                  LD      HL,swR+3
169 30A3' C4 AA 34                  CALL    NZ,amatch
170 30A6' 20 06                     JR      NZ,CC26         ;'/R' Command
171 30A8'
172 30A8' CD 94 32                  CALL    putout
173 30AB' C3 FA 1F                  JP      _HOT
174 30AE'
175 30AE' 21 FD 30          CC26:   LD      HL,swK
176 30B1' CD AA 34                  CALL    amatch
177 30B4' 21 00 31                  LD      HL,swK+3
178 30B7' C4 AA 34                  CALL    NZ,amatch
179 30BA' 20 06                     JR      NZ,CC31         ;'/K' Command
180 30BC'
181 30BC' CD 2A 32                  CALL    kill
182 30BF' C3 C8 30                  JP      CC12
183 30C2'
184 30C2' CD 11 31          CC31:   CALL    GETREL
185 30C5' DC 33 20                  CALL    C,_ERROR
186 30C8'
187 30C8' FD 7E 00          CC12:   LD      A,(IY)
188 30CB' A7                        AND     A
189 30CC' 28 14                     JR      Z,CC5
190 30CE' FE 2C                     CP      ','
191 30D0' FD 23                     INC     IY
192 30D2' 28 0E                     JR      Z,CC5
193 30D4'
194 30D4' 21 0B 31                  LD      HL,strERR
195 30D7' CD B1 35                  CALL    _puts
196 30DA' FD 2A 76 1F               LD      IY,(_KBFAD)
197 30DE' FD 36 00 00               LD      (IY),0
198 30E2'
199 30E2' C3 42 30          CC5:    JP      CC8
200 30E5'
201 30E5'
202 30E5' 2F 43 00 2F 63    swC:    DB      '/C',0,'/c',0
203 30EA' 00
204 30EB' 2F 45 00 2F 65    swE:    DB      '/E',0,'/e',0
205 30F0' 00
206 30F1' 2F 4C 00 2F 6C    swL:    DB      '/L',0,'/l',0
207 30F6' 00
208 30F7' 2F 52 00 2F 72    swR:    DB      '/R',0,'/r',0
209 30FC' 00
210 30FD' 2F 4B 00 2F 6B    swK:    DB      '/K',0,'/k',0
211 3102' 00
212 3103'
213 3103' 52 45 4C 00       strREL: DB      'REL',0
214 3107' 4C 49 42 00       strLIB: DB      'LIB',0
215 310B' 3F 45 72 72 0D    strERR: DB      '?Err',CR,0
216 3110' 00
217 3111'
218 3111'                   ;
219 3111'                   ; 'REL' File ﾉ ｼｮﾘ
220 3111'                   ;
221 3111' 21 ** **          GETREL: LD      HL,inname
222 3114' CD D8 34                  CALL    there           ;if(there(inname) == NO)
223 3117' 11 ** **                  LD      DE,inname       ;       flneat(inname."REL")
224 311A' 21 03 31                  LD      HL,strREL
225 311D' DC E2 34                  CALL    C,flneat
226 3120'
227 3120' 3E 01                     LD      A,1             ;Bin File
228 3122' 11 ** **                  LD      DE,inname
229 3125' CD A3 1F                  CALL    _FILE
230 3128' D8                        RET     C
231 3129' CD 09 20                  CALL    _ROPEN
232 312C' 38 21                     JR      C,CC36          ;'....REL' ｶﾞ ﾅｹﾚﾊﾞ '....LIB' ｦ ｻｶﾞｽ
233 312E'
234 312E' DD 36 00 01               LD      (IX),REL        ; ｲｶ 'REL' file ﾉ ｼｮﾘ
235 3132' DD 23                     INC     IX
236 3134' 2A 72 1F                  LD      HL,(_SIZE)
237 3137' DD 75 00                  LD      (IX),L
238 313A' DD 23                     INC     IX
239 313C' DD 74 00                  LD      (IX),H
240 313F' DD 23                     INC     IX
241 3141'
242 3141' DD E5                     PUSH    IX
243 3143' D1                        POP     DE              ;LD DE,IX
244 3144' 21 ** **                  LD      HL,inname
245 3147' CD FC 31                  CALL    fcopy
246 314A' E5                        PUSH    HL
247 314B' DD E1                     POP     IX              ;LD IX,HL
248 314D' B7                        OR      A               ;CY = 0
249 314E' C9                        RET
250 314F'
251 314F'                   ;
252 314F'                   ; LIB File ﾉ ｼｮﾘ
```

THE SENTINEL

▶このおれにアドベンチャーの素晴らしさを教えてくれた清水和人氏がついに帰ってきた。そこで清水氏に復活を祝して北斗賞を送ろう。　古岡　伸一 (19) 山口県

```
253  314F'                     ;
254  314F' FE 08               CC36:  CP     8               ;File not found ﾀﾞｹ ｺﾙｽ
255  3151' 37                         SCF
256  3152' C0                         RET    NZ
257  3153'
258  3153' 11 ** **                   LD     DE,inname
259  3156' 21 07 31                   LD     HL,strLIB
260  3159' CD E2 34                   CALL   flneat          ;flneat(inname,"LIB")
261  315C'
262  315C' 3E 01                      LD     A,1             ;Bin File
263  315E' 11 ** **                   LD     DE,inname
264  3161' CD ** **                   CALL   RDOPEN
265  3164' D8                         RET    C
266  3165'
267  3165' DD 36 00 02                LD     (IX),LIB
268  3169' DD 23                      INC    IX
269  316B'
270  316B' DD E5                      PUSH   IX
271  316D' D1                         POP    DE              ;LD DE,IX
272  316E' 21 ** **                   LD     HL,inname
273  3171' CD FC 34                   CALL   fcopy
274  3174' E5                         PUSH   HL
275  3175' DD E1                      POP    IX              ;LD IX,HL
276  3177'
277  3177' 21 00 00                   LD     HL,0
278  317A' 22 ** **                   LD     (LIBseek),HL
279  317D' 21 ** **                   LD     HL,DUMMY - 1
280  3180' 22 ** **                   LD     (LIBadrs),HL    ;Cancel first Adjustment
281  3183'
282  3183' CD ** **            CC50:  CALL   INPUT_
283  3186' FE FF                      CP     0FFH
284  3188' CA CD 31                   JP     Z,CC52
285  318B' FE E0                      CP     0E0H
286  318D' C2 25 35                   JP     NZ,ERR
287  3190'
288  3190' DD 36 00 03                LD     (IX),MOD
289  3194' CD ** **                   CALL   INPUT_
290  3197' DD 77 03                   LD     (IX+3),A
291  319A' CD ** **                   CALL   INPUT_
292  319D' DD 77 04                   LD     (IX+4),A        ;Seek Addres for input
293  31A0'
294  31A0' 67                         LD     H,A
295  31A1' DD 6E 03                   LD     L,(IX+3)        ;HL = seek Address for input
296  31A4' ED 5B ** **                LD     DE,(LIBseek)
297  31A8' 22 ** **                   LD     (LIBseek),HL
298  31AB' B7                         OR     A
299  31AC' ED 52                      SBC    HL,DE
300  31AE' EB                         EX     DE,HL           ;DE = new (LIBseek) - old (LIBseek)
301  31AF' 2A ** **                   LD     HL,(LIBadrs)
302  31B2' 23                         INC    HL
303  31B3' 73                         LD     (HL),E
304  31B4' 23                         INC    HL
305  31B5' 72                         LD     (HL),D
306  31B6'
307  31B6' DD 22 ** **                LD     (LIBadrs),IX
308  31BA'
309  31BA' 11 05 00                   LD     DE,5
310  31BD' DD 19                      ADD    IX,DE           ;IX=IX+5
311  31BF'
312  31BF' CD ** **            CC51:  CALL   INPUT_
313  31C2' DD 77 00                   LD     (IX),A
314  31C5' DD 23                      INC    IX
315  31C7' A7                         AND    A
316  31C8' 20 F5                      JR     NZ,CC51
317  31CA' C3 83 31                   JP     CC50
318  31CD'
319  31CD' 2A ** **            CC52:  LD     HL,(wkin+12H)
320  31D0' ED 5B ** **                LD     DE,(LIBseek)
321  31D4' B7                         OR     A
322  31D5' ED 52                      SBC    HL,DE
323  31D7' EB                         EX     DE,HL           ;DE = Last Modlue Size
324  31D8' 2A ** **                   LD     HL,(LIBadrs)
325  31DB' 23                         INC    HL
326  31DC' 73                         LD     (HL),E
327  31DD' 23                         INC    HL
328  31DE' 72                         LD     (HL),D
329  31DF' B7                         OR     A               ;CY = 0
330  31E0' C9                         RET
331  31E1'
332  31E1'                     ;
333  31E1'                     ; Print out Entried Modlule Name
334  31E1'                     ;
335  31E1' DD 36 00 FF         prtmod: LD    (IX),EOF
336  31E5' 21 00 50                   LD     HL,cmdbuf
337  31E8' 7E                  prt1:  LD     A,(HL)
338  31E9' 23                         INC    HL
339  31EA' FE FF                      CP     EOF
340  31EC' CA EB 1F                   JP     Z,_NL
341  31EF'
342  31EF' FE 01                      CP     REL
343  31F1' CC 16 32                   CALL   Z,pmod2
344  31F4' FE 02                      CP     LIB
345  31F6' CC 0E 32                   CALL   Z,umod0
346  31F9' FE 03                      CP     MOD
347  31FB' CC 14 32                   CALL   Z,pmod4
348  31FE' FE 04                      CP     UREL
349  3200' CC 0C 32                   CALL   Z,umod2
350  3203' FE 05                      CP     UMOD
351  3205' CC 0A 32                   CALL   Z,umod4
352  3208' 18 DE                      JR     prt1
353  320A'
354  320A'
355  320A' 23                  umod4: INC    HL
356  320B' 23                         INC    HL
357  320C' 23                  umod2: INC    HL
358  320D' 23                         INC    HL
359  320E' 7E                  umod0: LD     A,(HL)
360  320F' 23                         INC    HL
361  3210' A7                         AND    A
362  3211' 20 FB                      JR     NZ,umod0
363  3213' C9                         RET
364  3214'
365  3214' 23                  pmod4: INC    HL
366  3215' 23                         INC    HL
367  3216' 23                  pmod2: INC    HL
368  3217' 23                         INC    HL
369  3218' CD B1 35                   CALL   _puts
370  321B'                     ;      CALL   prtTAB
371  321B'                     ;      RET
372  321B'
373  321B'                     ; Print TAB code
374  321B' D5                  prtTAB: PUSH  DE
375  321C' ED 5B 7A 1F                LD     DE,(_PRCNT)
376  3220' 1A                         LD     A,(DE)
377  3221' D1                         POP    DE
378  3222' E6 07                      AND    7
379  3224' C8                         RET    Z
380  3225' CD F1 1F                   CALL   _PRNTS
381  3228' 18 F1                      JR     prtTAB
382  322A'
383  322A'                     ;
384  322A'                     ; Kill the Entried Modlule Name
385  322A'                     ;
386  322A' DD 36 00 FF         kill:  LD     (IX),EOF
387  322E' 21 00 50                   LD     HL,cmdbuf
388  3231' 22 ** **            kil1:  LD     (HLBUF),HL
389  3234' 7E                         LD     A,(HL)
390  3235' 23                         INC    HL
391  3236' FE FF                      CP     EOF
392  3238' 28 2B                      JR     Z,kil1          ;Not Found
393  323A'
394  323A' FE 01                      CP     REL             ;if (HL)=REL ■
395  323C' CC 7A 32                   CALL   Z,ckil2         ;   if SAME goto kil2
396  323F' 28 18                      JR     Z,kil2          ;   }
397  3241' FE 02                      CP     LIB
398  3243' CC 0E 32                   CALL   Z,umod0
399  3246' FE 03                      CP     MOD             ;if (HL)=MOD ■
400  3248' CC 78 32                   CALL   Z,ckil4         ;   if SAME goto kil3
401  324B' 28 12                      JR     Z,kil3          ;   }
402  324D' FE 04                      CP     UREL
403  324F' CC 0C 32                   CALL   Z,umod2
404  3252' FE 05                      CP     UMOD
405  3254' CC 0A 32                   CALL   Z,umod4
406  3257' 18 D8                      JR     kil1
407  3259'
408  3259' 2A ** **            kil2:  LD     HL,(HLBUF)
409  325C' 36 04                      LD     (HL),UREL
410  325E' C9                         RET
411  325F'
412  325F' 2A ** **            kil3:  LD     HL,(HLBUF)
413  3262' 36 05                      LD     (HL),UMOD
414  3264' C9                         RET
415  3265'
416  3265' 21 6C 32            kil4:  LD     HL,kil5
417  3268' CD B1 35                   CALL   _puts
418  326B' C9                         RET
419  326C'
420  326C' 4E 6F 74 20 46      kil5:  DB     'Not Found ',CR,0
421  3271' 6F 75 6E 64 20
422  3276' 0D 00
423  3278'
424  3278' 23                  ckil4: INC    HL
425  3279' 23                         INC    HL
426  327A' 23                  ckil2: INC    HL
427  327B' 23                         INC    HL
428  327C' 11 ** **            ckil0: LD     DE,inname
429  327F' 1A                  kki1:  LD     A,(DE)
430  3280' A7                         AND    A
431  3281' 28 07                      JR     Z,kki2
432  3283' BE                         CP     (HL)
433  3284' 20 07                      JR     NZ,kki3
434  3286' 13                         INC    DE
435  3287' 23                         INC    HL
436  3288' 18 F5                      JR     kki1
437  328A'
438  328A' 7E                  kki2:  LD     A,(HL)
439  328B' A7                         AND    A
440  328C' C9                         RET                    ;Z=0 Not-Matched ,Z=1 Matched
441  328D'
442  328D' CD 0E 32            kki3:  CALL   umod0
443  3290' 3E 01                      LD     A,1
444  3292' A7                         AND    A               ;Z=0 Not-matched
445  3293' C9                         RET
446  3294'
447  3294'                     ;
448  3294'                     ; 'LIB' File ﾉ ｼｭﾂﾘｮｸ
449  3294'                     ;
450  3294' DD 36 00 FF         putout: LD    (IX),EOF        ;Put End code
451  3298'
452  3298' 11 ** **                   LD     DE,outname
453  329B' 21 ** **                   LD     HL,inname
454  329E' CD FC 34                   CALL   fcopy
455  32A1'
456  32A1' 21 ** **                   LD     HL,outname
457  32A4' CD D8 34                   CALL   there
458  32A7' 11 ** **                   LD     DE,outname
459  32AA' 21 07 31                   LD     HL,strLIB
460  32AD' DC E2 34                   CALL   C,flneat
461  32B0'
462  32B0' 3E 01                      LD     A,1
463  32B2' 11 ** **                   LD     DE,outname
464  32B5' CD A3 1F                   CALL   _FILE
465  32B8' D8                         RET    C
466  32B9' 3A 5D 1F                   LD     A,(_DSK)        ;A = Device Number
467  32BC' 32 D8 35                   LD     (tmpname),A
468  32BF'
469  32BF' 3E 01                      LD     A,1             ;Bin File
470  32C1' 11 D8 35                   LD     DE,tmpname
471  32C4' CD ** **                   CALL   WROPEN
472  32C7' D8                         RET    C
473  32C8'
474  32C8' CD F5 32                   CALL   counthd
475  32CB' 22 ** **                   LD     (HDSIZE),HL
476  32CE' CD 5F 33                   CALL   makehd
477  32D1' CD F5 33                   CALL   makelib
478  32D4'
479  32D4' CD ** **                   CALL   CLOSE           ;close tmp file
480  32D7' D8                         RET    C
481  32D8'
482  32D8' 11 ** **                   LD     DE,outname
483  32DB' CD A3 1F                   CALL   _FILE
484  32DE' D8                         RET    C
485  32DF' CD ** **                   CALL   FCBSCH          ;FCB Search
486  32E2' D8                         RET    C
487  32E3' CC 15 20                   CALL   Z,_KILL         ;Kill old outname if exsits
488  32E6' D8                         RET    C
489  32E7'
490  32E7' 11 D8 35                   LD     DE,tmpname      ;rename tmp-file to outname
491  32EA' CD A3 1F                   CALL   _FILE
492  32ED' D8                         RET    C
493  32EE' 11 ** **                   LD     DE,outname
494  32F1' CD 12 20                   CALL   _NAME
495  32F4' C9                         RET
496  32F5'
497  32F5'                     ;
498  32F5'                     ; Count File Header Size
499  32F5'                     ;
500  32F5' DD 21 00 50         counthd:LD    IX,cmdbuf
501  32F9' 21 01 00                   LD     HL,1            ;There is always EOF Code
502  32FC' DD 7E 00            CC90:  LD     A,(IX)
503  32FF' DD 23                      INC    IX
504  3301' FE FF                      CP     EOF
505  3303' C8                         RET    Z
506  3304'
507  3304' FE 01                      CP     REL
508  3306' 20 0C                      JR     NZ,CC91
509  3308'
510  3308' DD 23                      INC    IX
511  330A' DD 23                      INC    IX
512  330C' 23                         INC    HL              ;E0-ITEM
513  330D' 23                         INC    HL              ;Offset Address
514  330E' 23                         INC    HL
515  330F' CD 43 33                   CALL   acstr
516  3312' 18 E8                      JR     CC90
517  3314'
518  3314' FE 02               CC91:  CP     LIB
519  3316' 20 05                      JR     NZ,CC92
520  3318'
521  3318' CD A1 34                   CALL   spstr
522  331B' 18 DF                      JR     CC90
523  331D'
524  331D' FE 03               CC92:  CP     MOD
525  331F' 20 10                      JR     NZ,CC93
526  3321'
527  3321' DD 23                      INC    IX
528  3323' DD 23                      INC    IX
529  3325' DD 23                      INC    IX
530  3327' DD 23                      INC    IX
531  3329' 23                         INC    HL              ;E0-ITEM
532  332A' 23                         INC    HL              ;Offset Address
533  332B' 23                         INC    HL
534  332C' CD 43 33                   CALL   acstr
535  332F' 18 CB                      JR     CC90
536  3331'
537  3331' FE 04               CC93:  CP     UREL
538  3333' 28 04                      JR     Z,CC94
539  3335'
540  3335' DD 23                      INC    IX              ;UMOD
541  3337' DD 23                      INC    IX
542  3339' DD 23               CC94:  INC    IX
```


▶最近よく思うことなんだけど，小さい頃は早く大人になりたいと思ったけど，大きくなるにつれて，子供に戻りたいと思うようになってきた。小学校の頃は楽しかった。外で1日中遊んでたもんなぁ。これを読んでる小学生の人(ゼロに近いと思うけど)，ファミコンばっかしてんと外で思いっきり遊びなさい。　　松嶋　祥文 (17) 京都府

```
543  333B' DD 23                   INC     IX
544  333D' CD A1 34                CALL    spstr
545  3340' C3 FC 32                JP      CC90
546  3343'
547  3343'                 ; Add HL with count of string(file name)
548  3343'
549  3343' DD 7E 01        acstr:  LD      A,(IX+1)
550  3346' FE 3A                   CP      ':'
551  3348' 20 04                   JR      NZ,acstr1
552  334A' DD 23                   INC     IX
553  334C' DD 23                   INC     IX              ;Skip Device Name
554  334E' DD 7E 00        acstr1: LD      A,(IX)
555  3351' DD 23                   INC     IX
556  3353' 23                      INC     HL
557  3354' FE 2E                   CP      '.'
558  3356' CA A1 34                JP      Z,spstr         ;Skip EXT-name
559  3359' CD 78 35                CALL    isflchr
560  335C' 20 F0                   JR      NZ,acstr1
561  335E' C9                      RET
562  335F'
563  335F'                 ;
564  335F'                 ; Making File Header
565  335F'                 ;
566  335F' DD 21 00 50     makehd: LD      IX,cmdbuf
567  3363' 2A ** **                LD      HL,(HDSIZE)
568  3366' 22 ** **                LD      (flptr),HL
569  3369'
570  3369' DD 7E 00        CC60:   LD      A,(IX)
571  336C' DD 23                   INC     IX
572  336E' FE FF                   CP      EOF
573  3370' CA 84 34                JP      Z,puteof
574  3373'
575  3373' FE 01                   CP      REL
576  3375' 20 32                   JR      NZ,CC62         ;REL Command ノ ショリ
577  3377'
578  3377' 3E E0                   LD      A,0E0H          ;MOD ITEM
579  3379' CD ** **                CALL    PRINT_
580  337C'
581  337C' 3A ** **                LD      A,(flptr)
582  337F' CD ** **                CALL    PRINT_
583  3382' 3A ** **                LD      A,(flptr+1)
584  3385' CD ** **                CALL    PRINT_
585  3388'
586  3388' 2A ** **                LD      HL,(flptr)
587  338B' DD 5E 00                LD      E,(IX)
588  338E' DD 23                   INC     IX
589  3390' DD 56 00                LD      D,(IX)
590  3393' DD 23                   INC     IX
591  3395' 19                      ADD     HL,DE
592  3396' 22 ** **                LD      (flptr),HL
593  3399'
594  3399' DD 7E 01                LD      A,(IX+1)
595  339C' FE 3A                   CP      ':'
596  339E' 20 04                   JR      NZ,CC61
597  33A0'
598  33A0' DD 23                   INC     IX
599  33A2' DD 23                   INC     IX              ;Skip Device Name
600  33A4'
601  33A4' CD BC 34        CC61:   CALL    putmod          ;Put Module-Name
602  33A7' 18 C0                   JR      CC60
603  33A9'
604  33A9' FE 02           CC62:   CP      LIB             ;LIB Command ノ ショリ
605  33AB' 20 06                   JR      NZ,CC64
606  33AD'
607  33AD' CD A1 34                CALL    spstr           ;Cancel thie command
608  33B0' C3 69 33                JP      CC60
609  33B3'
610  33B3' FE 03           CC64:   CP      MOD
611  33B5' 20 2C                   JR      NZ,CC66
612  33B7'
613  33B7' 3E E0                   LD      A,0E0H
614  33B9' CD ** **                CALL    PRINT_          ;MOD Command ノ ショリ
615  33BC'
616  33BC' 3A ** **                LD      A,(flptr)
617  33BF' CD ** **                CALL    PRINT_
618  33C2' 3A ** **                LD      A,(flptr+1)
619  33C5' CD ** **                CALL    PRINT_
620  33C8'
621  33C8' 2A ** **                LD      HL,(flptr)
622  33CB' DD 5E 00                LD      E,(IX)          ;MOD Command ノ ショリ
623  33CE' DD 23                   INC     IX
624  33D0' DD 56 00                LD      D,(IX)
625  33D3' DD 23                   INC     IX
626  33D5' 19                      ADD     HL,DE
627  33D6' 22 ** **                LD      (flptr),HL
628  33D9' DD 23                   INC     IX
629  33DB' DD 23                   INC     IX              ;Skip Seek-Address for input
630  33DD' CD BC 34                CALL    putmod          ;Put Module name
631  33E0' C3 69 33                JP      CC60
632  33E3'
633  33E3' FE 04           CC66:   CP      UREL
634  33E5' 28 04                   JR      Z,CC67
635  33E7'
636  33E7' DD 23                   INC     IX              ;UMOD
637  33E9' DD 23                   INC     IX
638  33EB' DD 23           CC67:   INC     IX
639  33ED' DD 23                   INC     IX
640  33EF' CD A1 34                CALL    spstr
641  33F2' C3 69 33                JP      CC60
642  33F5'
643  33F5'
644  33F5'                 ;
645  33F5'                 ; Make Libraru file
646  33F5'                 ;
647  33F5' DD 21 00 50     makelib:LD      IX,cmdbuf
648  33F9' DD 7E 00        CC70:   LD      A,(IX)
649  33FC' DD 23                   INC     IX
650  33FE' FE FF                   CP      EOF
651  3400' C8                      RET     Z               ;RET with CY = 0
652  3401'
653  3401' FE 01                   CP      REL
654  3403' 20 32                   JR      NZ,CC72
655  3405'
656  3405' DD 4E 00                LD      C,(IX)
657  3408' DD 23                   INC     IX
658  340A' DD 46 00                LD      B,(IX)
659  340D' DD 23                   INC     IX              ;BC = (File size)
660  340F' C5                      PUSH    BC
661  3410'
662  3410' DD E5                   PUSH    IX
663  3412' E1                      POP     HL              ;LD HL,IX
664  3413' 11 ** **                LD      DE,inname
665  3416' CD 13 35                CALL    fcopy2
666  3419' 23                      INC     HL              ;Skip End-code of File-Name
667  341A' E5                      PUSH    HL
668  341B' DD E1                   POP     IX              ;LD IX,HL
669  341D'
670  341D' 3E 01                   LD      A,1             ;Bin File
671  341F' 11 ** **                LD      DE,inname
672  3422' CD ** **                CALL    RDOPEN
673  3425' C1                      POP     BC
674  3426' D8                      RET     C               ;Read Error
675  3427'
676  3427' C5              CC71:   PUSH    BC
677  3428' CD ** **                CALL    INPUT_
678  342B' CD ** **                CALL    PRINT_
679  342E' C1                      POP     BC
680  342F' 0B                      DEC     BC
681  3430' 78                      LD      A,B
682  3431' B1                      OR      C
683  3432' 20 F3                   JR      NZ,CC71
684  3434' C3 F9 33                JP      CC70
685  3437'
686  3437' FE 02           CC72:   CP      LIB
687  3439' 20 19                   JR      NZ,CC73
688  343B'
689  343B' DD E5                   PUSH    IX
690  343D' E1                      POP     HL              ;LD HL,IX
691  343E' 11 ** **                LD      DE,inname
692  3441' CD 13 35                CALL    fcopy2
693  3444' 23                      INC     HL              ;Skip End-Code of File-Name
694  3445' E5                      PUSH    HL
695  3446' DD E1                   POP     IX              ;LD IX,HL
696  3448'
697  3448' 3E 01                   LD      A,1             ;Bin File
698  344A' 11 ** **                LD      DE,inname
699  344D' CD ** **                CALL    RDOPEN
700  3450' D8                      RET     C               ;Read Error
701  3451' C3 F9 33                JP      CC70
702  3454'
703  3454' FE 03           CC73:   CP      MOD
704  3456' 20 1C                   JR      NZ,CC75
705  3458'
706  3458' DD 6E 02                LD      L,(IX+2)
707  345B' DD 66 03                LD      H,(IX+3)
708  345E' CD ** **                CALL    FSEEK
709  3461' DD 4E 00                LD      C,(IX)
710  3464' DD 46 01                LD      B,(IX+1)
711  3467' C5              CC74:   PUSH    BC              ;BC = Module Size
712  3468' CD ** **                CALL    INPUT_
713  346B' CD ** **                CALL    PRINT_
714  346E' C1                      POP     BC
715  346F' 0B                      DEC     BC
716  3470' 78                      LD      A,B
717  3471' B1                      OR      C
718  3472' 20 F3                   JR      NZ,CC74
719  3474'
720  3474'                 ;       INC     IX
721  3474'                 ;       INC     IX
722  3474'                 ;       INC     IX
723  3474'                 ;       INC     IX
724  3474'                 ;       CALL    spstr
725  3474'                 ;       RET
726  3474'
727  3474' FE 04           CC75:   CP      UREL
728  3476' 28 04                   JR      Z,CC76
729  3478'
730  3478'                 ; イカ UMOD Command ノ ショリ
731  3478' DD 23                   INC     IX
732  347A' DD 23                   INC     IX
733  347C' DD 23           CC76:   INC     IX
734  347E' DD 23                   INC     IX
735  3480' CD A1 34                CALL    spstr
736  3483' C9                      RET
737  3484'
738  3484'
739  3484'
740  3484'
741  3484' 3E FF           puteof: LD      A,0FFH
742  3486' CD ** **                CALL    PRINT_
743  3489' C9                      RET
744  348A'
745  348A'                 ;
746  348A'                 ; Subroutines
747  348A'                 ;
748  348A'
749  348A'                 ; Get Line input From console
750  348A'
751  348A' 3E 2A           GETL:   LD      A,'*'
752  348C' CD F4 1F                CALL    _PRINT          ;Print Prompt
753  348F' ED 5B 76 1F             LD      DE,(_KBFAD)
754  3493' CD D3 1F                CALL    _GETL
755  3496' 1A                      LD      A,(DE)
756  3497' FE 1B                   CP      BRK
757  3499' CA FA 1F                JP      Z,_HOT
758  349C' 13                      INC     DE              ;Skip Prompt '*'
759  349D' D5                      PUSH    DE
760  349E' FD E1                   POP     IY              ;LD IY,DE
761  34A0' C9                      RET
762  34A1'
763  34A1'                 ; Skip String
764  34A1'
765  34A1' DD 7E 00        spstr:  LD      A,(IX)
766  34A4' DD 23                   INC     IX
767  34A6' A7                      AND     A
768  34A7' 20 F8                   JR      NZ,spstr
769  34A9' C9                      RET
770  34AA'
771  34AA'                 ; Compare Strings
772  34AA'
773  34AA' FD E5           amatch: PUSH    IY
774  34AC' D1                      POP     DE              ;LD DE,IY
775  34AD' 7E              CC88:   LD      A,(HL)
776  34AE' A7                      AND     A
777  34AF' 28 07                   JR      Z,CC89
778  34B1' 1A                      LD      A,(DE)
779  34B2' BE                      CP      (HL)
780  34B3' C0                      RET     NZ              ;Not Matched
781  34B4' 13                      INC     DE
782  34B5' 23                      INC     HL
783  34B6' 18 F5                   JR      CC88
784  34B8'
785  34B8' D5              CC89:   PUSH    DE
786  34B9' FD E1                   POP     IY              ;LD IY,DE
787  34BB' C9                      RET                     ;Matched
788  34BC'
789  34BC'
790  34BC'                 ; Put Module Name into file
791  34BC'
792  34BC' DD 7E 00        putmod: LD      A,(IX)
793  34BF' FE 2E                   CP      '.'
794  34C1' 28 0C                   JR      Z,putmod1       ;Skip EXT-name
795  34C3' CD ** **                CALL    PRINT_
796  34C6' DD 7E 00                LD      A,(IX)
797  34C9' DD 23                   INC     IX
798  34CB' A7                      AND     A
799  34CC' C8                      RET     Z
800  34CD' 18 ED                   JR      putmod
801  34CF'
802  34CF' CD A1 34        putmod1:CALL    spstr
803  34D2' 3E 00                   LD      A,0
804  34D4' CD ** **                CALL    PRINT_
805  34D7' C9                      RET
806  34D8'
807  34D8'
808  34D8'                 ; Wether there is '.'
809  34D8'
810  34D8' 7E              there:  LD      A,(HL)
811  34D9' 23                      INC     HL
812  34DA' A7                      AND     A
813  34DB' 37                      SCF
814  34DC' C8                      RET     Z               ;RET With CY = 1
815  34DD' FE 2E                   CP      '.'
816  34DF' 20 F7                   JR      NZ,there
817  34E1' C9                      RET                     ;RET With CY = 0
818  34E2'
819  34E2' 06 0F           flneat: LD      B,NAME          ;B = Counter
820  34E4' 1A              CC97:   LD      A,(DE)
821  34E5' CD 78 35                CALL    isflchr
822  34E8' 38 0A                   JR      C,CC98
823  34EA' 05                      DEC     B
824  34EB' 28 07                   JR      Z,CC98
825  34ED'
826  34ED' FE 2E                   CP      '.'
827  34EF' 28 03                   JR      Z,CC98
828  34F1' 13                      INC     DE
829  34F2' 18 F0                   JR      CC97
830  34F4'
831  34F4' 3E 2E           CC98:   LD      A,'.'
832  34F6' 12                      LD      (DE),A
```


▶メガネを変えたら星が見えた。でも星の数が少ないのは，目の悪いせいだけではないようだ。
森田　泰史（17）埼玉県

```
833  34F7' 13                        INC     DE
834  34F8' CD FC 34                  CALL    fcopy
835  34FB' C9                        RET
836  34FC'
837  34FC' 06 14             fcopy:  LD      B,LENFL
838  34FE'
839  34FE' AF                        XOR     A
840  34FF' 12                        LD      (DE),A
841  3500'
842  3500' 7E                CC104:  LD      A,(HL)
843  3501' 23                        INC     HL
844  3502' CD 78 35                  CALL    isflchr
845  3505' 38 09                     JR      C,CC105
846  3507' 05                        DEC     B
847  3508' 28 06                     JR      Z,CC105
848  350A'
849  350A' 12                        LD      (DE),A
850  350B' 13                        INC     DE
851  350C' AF                        XOR     A
852  350D' 12                        LD      (DE),A
853  350E' 18 F0                     JR      CC104
854  3510'
855  3510' EB                CC105:  EX      DE,HL
856  3511' 23                        INC     HL
857  3512' C9                        RET
858  3513'
859  3513' 06 14             fcopy2: LD      B,LENFL
860  3515' AF                        XOR     A
861  3516' 12                        LD      (DE),A
862  3517' 7E                CC109:  LD      A,(HL)
863  3518' CD 78 35                  CALL    isflchr
864  351B' D8                        RET     C
865  351C' 05                        DEC     B
866  351D' C8                        RET     Z
867  351E'
868  351E' 23                        INC     HL
869  351F' 12                        LD      (DE),A
870  3520' 13                        INC     DE
871  3521' AF                        XOR     A
872  3522' 12                        LD      (DE),A
873  3523' 18 F2                     JR      CC109
874  3525'
875  3525'
876  3525' 21 4B 35          ERR:  LD        HL,EMSG2
877  3528' CD B1 35                CALL      _puts
878  352B' 21 ** **                LD        HL,inname
879  352E' CD B1 35                CALL      _puts
880  3531' CD EE 1F                CALL      _LTNL
881  3534' C9                      RET
882  3535'
883  3535'
884  3535'
885  3535'                   ;
886  3535'                   ; Error Messages
887  3535'                   ;
888  3535' 46 69 6C 65 20    EMSG1:  DB      'File Access Error on:',0
889  353A' 41 63 63 65 73
890  353F' 73 20 45 72 72
891  3544' 6F 72 20 6F 6E
892  3549' 3A 00
893  354B' 49 6C 6C 65 67    EMSG2:  DB      'Illegal file ',CR,0
894  3550' 61 6C 20 66 69
895  3555' 6C 65 20 0D 00
896  355A'
897  355A' 42 61 64 20 43    EMSG3:  DB      'Bad Command ',CR,0
898  355F' 6F 6D 6D 61 6E
899  3564' 64 20 0D 00
900  3568' 46 69 6C 65 20    EMSG4:  DB      'File not found',CR,0
901  356D' 6E 6F 74 20 66
902  3572' 6F 75 6E 64 0D
903  3577' 00
904  3578'
905  3578'
906  3578' FE 2F             isflchr:CP      '/'
907  357A' 28 06                     JR      Z,isfl1
908  357C' FE 2C                     CP      ','
909  357E' 28 02                     JR      Z,isfl1
910  3580' A7                        AND     A
911  3581' C0                        RET     NZ              ;CY = 0
912  3582' 37                isfl1:  SCF
913  3583' C9                        RET                     ;CY = 1
914  3584'
915  3584' 21 00 00          hlhex:  LD      HL,0
916  3587'
917  3587' FD 7E 00          CC118:  LD      A,(IY)
918  358A' 16 30                     LD      D,'0'
919  358C' FE 30                     CP      '0'
920  358E' D8                        RET     C
921  358F' FE 3A                     CP      '9' + 1
922  3591' 38 11                     JR      C,CC119
923  3593'
924  3593' 16 41                     LD      D,'A'
925  3595' FE 41                     CP      'A'
926  3597' D8                        RET     C
927  3598' FE 47                     CP      'F' + 1
928  359A' 38 08                     JR      C,CC119
929  359C'
930  359C' 16 61                     LD      D,'a'
931  359E' FE 61                     CP      'a'
932  35A0' D8                        RET     C
933  35A1' FE 67                     CP      'f' + 1
934  35A3' D0                        RET     NC
935  35A4'
936  35A4' 29                CC119:  ADD     HL,HL
937  35A5' 29                        ADD     HL,HL
938  35A6' 29                        ADD     HL,HL
939  35A7' 29                        ADD     HL,HL           ;HL = HL * 16
940  35A8' 92                        SUB     D
941  35A9' 16 00                     LD      D,0
942  35AB' 5F                        LD      E,A
943  35AC' 19                        ADD     HL,DE
944  35AD' FD 23                     INC     IY
945  35AF' 18 D6                     JR      CC118
946  35B1'
947  35B1'
948  35B1' 7E                _puts:  LD      A,(HL)
949  35B2' 23                        INC     HL
950  35B3' A7                        AND     A
951  35B4' C8                        RET     Z
952  35B5' CD F4 1F                  CALL    _PRINT
953  35B8' 18 F7                     JR      _puts
954  35BA'
955  35BA'
956  35BA' 57 4C 42 20 76    TITLE:  DB      'WLB ver 1.00 (by T.Ishigami)',CR,0
957  35BF' 65 72 20 31 2E
958  35C4' 30 30 20 28 62
959  35C9' 79 20 54 2E 49
960  35CE' 73 68 69 67 61
961  35D3' 6D 69 29 0D 00
962  35D8'
963  35D8'
964  35D8'                   ; Temporary File name
965  35D8' 41 3A 57 4C 42    tmpname:DB      'A:WLB          .TMP',0
966  35DD' 20 20 20 20 20
967  35E2' 20 20 20 20 20
968  35E7' 2E 54 4D 50 00
969  35EC'
970  35EC'
971  35EC'
972  35EC'                   ;
973  35EC'                   ; Works
974  35EC'                   ;
975  35EC'                           DSEG
976  0000"
977  0000"
978  0000"                   inname: DS      LENFL           ;input  file name
979  0014"                   outname:DS      LENFL           ;output file name
980  0028"                   HLBUF:  DS      2               ;Stack for HL reg.
981  002A"                   LIBseek:DS      2               ;Stack for seekaddress
982  002C"                   LIBadrs:DS      2               ;Stack for address
983  002E"                   DUMMY:  DS      2               ;Dummy Work area
984  0030"                   HDSIZE: DS      2               ;File Header Size
985  0032"                   flptr:  DS      2               ;File Pointer
986  0034"
987  0034"                   ;
988  0034"                   ; Externals
989  0034"                   ;
990  0034"
991                                  EXT     WROPEN
992                                  EXT     RDOPEN
993                                  EXT     FSEEK
994                                  EXT     CLOSE
995                                  EXT     FCBSCH
996                                  EXT     INPUT_
997                                  EXT     PRINT_
998                                  EXT     wkin
999                                  EXT     wkout
1000 0034"                           END
```

リスト3 WLB2.ASM

```
  1  0000'                   ;**********************************
  2  0000'                   ; File Access Routine For WLB
  3  0000'                   ;       Programed by T.Ishigami
  4  0000'                   ;         '90 Feb.25th
  5  0000'                   ;
  6  0000'                   ;**********************************
  7  0000'
  8  0000'                           CSEG
  9  0000'
 10  0000'                 C         INCLUDE SOS.DEF
 11  1FFA                  C _HOT    EQU     1FFAH
 12  1FF4                  C _PRINT  EQU     1FF4H
 13  1FF1                  C _PRNTS  EQU     1FF1H
 14  1FEE                  C _LTNL   EQU     1FEEH
 15  1FEB                  C _NL     EQU     1FEBH
 16  1FE5                  C _MSX    EQU     1FE5H
 17  1FDF                  C _TAB    EQU     1FDFH
 18  1FD3                  C _GETL   EQU     1FD3H
 19  1FC7                  C _PAUSE  EQU     1FC7H
 20  1FC1                  C _PRTHX  EQU     1FC1H
 21  1FBE                  C _PRTHL  EQU     1FBEH
 22  1FB8                  C _HEX    EQU     1FB8H
 23  1FA3                  C _FILE   EQU     1FA3H
 24  1FAF                  C _WOPEN  EQU     1FAFH
 25  1F9A                  C _POKE   EQU     1F9AH
 26  1F94                  C _PEEK   EQU     1F94H
 27  2009                  C _ROPEN  EQU     2009H
 28  2015                  C _KILL   EQU     2015H
 29  2012                  C _NAME   EQU     2012H
 30  2033                  C _ERROR  EQU     2033H
 31  0000'                 C
 32  2000                  C _DRDSB  EQU     2000H
 33  2003                  C _DWTSB  EQU     2003H
 34  0000'                 C
 35  1F7A                  C _PRCNT  EQU     1F7AH
 36  1F76                  C _KBFAD  EQU     1F76H
 37  1F74                  C _IBFAD  EQU     1F74H
 38  1F72                  C _SIZE   EQU     1F72H
 39  1F6A                  C _MEMAX  EQU     1F6AH
 40  1F64                  C _DTBUF  EQU     1F64H
 41  1F62                  C _FATBF  EQU     1F62H
 42  1F60                  C _DIRPS  EQU     1F60H
 43  1F5E                  C _FATPOS EQU     1F5EH
 44  1F5D                  C _DSK    EQU     1F5DH
 45  0000'                 C
 46  0000'                 C         INCLUDE WLB.DEF
 47  0000'                 C ;================================
 48  0000'                 C ; WLB.DEF Header File For WLB
 49  0000'                 C ;     # WZD
 50  0000'                 C ;     #=WLB1
 51  0000'                 C ;     *=WLB2
 52  0000'                 C ;     # WLK /P:3000,/D:4500
 53  0000'                 C ;     * WLB1,WLB2,WLB/N:P
 54  0000'                 C ;     #
 55  0000'                 C ;     CSEG    3000H
 56  0000'                 C ;     DSEG    4500H
 57  0000'                 C ;
 58  0000'                 C ;================================
 59  0000'                 C
 60  B000                  C RDBUF   EQU     0B000H  ;- B0FFH
 61  B100                  C WRBUF   EQU     0B100H  ;- B1FFH
 62  0000'                 C
 63  5000                  C cmdbuf  EQU     5000H
 64  0000'                 C
 65  0000'
 66  0000'
 67  0000'                   ;====================
 68  0000'                   ; FILE OPEN FOR READ
 69  0000'                   ;====================
 70  0000' CD A3 1F          RDOPEN::CALL    _FILE
 71  0003' D8                        RET     C
 72  0004' CD ** **                  CALL    ROPEN
 73  0007' D8                        RET     C
 74  0008'                           ;
 75  0008' 2A 74 1F                  LD      HL,(_IBFAD)
 76  000B' 11 ** **                  LD      DE,FILE_BF
 77  000E' 01 20 00                  LD      BC,20H
 78  0011' ED B0                     LDIR
 79  0013' 3A 5D 1F                  LD      A,(_DSK)
 80  0016' 32 ** **                  LD      (FLDSK),A
 81  0019'
 82  0019' AF                        XOR     A
 83  001A' 32 ** **                  LD      (LST_DSK),A     ;NEVER BEING THE SAME
 84  001D'
 85  001D' CD ** **                  CALL    RDFAT
 86  0020' D8                        RET     C
 87  0021'
 88  0021' 06 10                     LD      B,10H
 89  0023' 0E 00                     LD      C,0             ; C <= TOTAL NUMBER OF CLUSTERS
 90  0025' 3A ** **                  LD      A,(FSTCLST)
 91  0028' 11 ** **                  LD      DE,TBLCLST
 92  002B'
 93  002B' 12                RDOPN4: LD      (DE),A
 94  002C' 13                        INC     DE
 95  002D' FE 7F                     CP      7FH
 96  002F' 30 0F                     JR      NC,RDOPN5
 97  0031' 2A 62 1F                  LD      HL,(_FATBF)
 98  0034' 85                        ADD     A,L
 99  0035' 6F                        LD      L,A
100  0036' 30 01                     JR      NC,RDOPN2
101  0038' 24                        INC     H
102  0039' 7E                RDOPN2: LD      A,(HL)          ;HL = (_FATBF) + (FSTCLST)
```


▶最近思うことは「大学に受かってバイトして，X68000を買う」ことだけ（18歳はだいたいこんなネタでしょう）。
前田　育男（18）新潟県

```
103  003A' 05                     DEC     B
104  003B' 28 28                  JR      Z,RDOPN3        ; More than 16 Cluster
105  003D' 0C                     INC     C
106  003E' 18 EB                  JR      RDOPN4
107  0040'
108  0040' 0D           RDOPN5: DEC     C               ;Last Cluster is Dummy
109  0041' F5                     PUSH    AF
110  0042' 79                     LD      A,C
111  0043' 87                     ADD     A,A
112  0044' 87                     ADD     A,A
113  0045' 87                     ADD     A,A
114  0046' 87                     ADD     A,A
115  0047' 4F                     LD      C,A             ;C = C * 16
116  0048' F1                     POP     AF
117  0049' D6 80                  SUB     80H             ;RC is 0 origin. So it isn't 7Fh but 80H.
118  004B' 81                     ADD     A,C
119  004C' 32 ** **               LD      (RC),A          ;RC = Total number of Records
120  004F'
121  004F' AF                     XOR     A
122  0050' 32 ** **               LD      (FLPNT),A
123  0053'
124  0053' 01 37 00               LD      BC,IMF_SIZE
125  0056' 11 ** **               LD      DE,wkin
126  0059' 21 ** **               LD      HL,FILE_BF
127  005C' ED B0                  LDIR
128  005E'
129  005E' 3E FF                  LD      A,0FFH
130  0060' 32 ** **               LD      (RDPNT),A       ;Clear Pointer For reading
131  0063'
132  0063' B7                     OR      A               ;CY = 0
133  0064' C9                     RET
134  0065'
135  0065' 3E 07        RDOPN3: LD      A,7             ;Bad Allocation Error
136  0067' 37                     SCF
137  0068' C9                     RET
138  0069'
139  0069'
140  0069' 3A 5D 1F     ROPEN:: LD      A,(_DSK)
141  006C' CD ** **               CALL    DEVCHK
142  006F' D8                     RET     C               ;Bad File descripter
143  0070' CD ** **               CALL    FCBSCH
144  0073' D8                     RET     C
145  0074' 3E 08                  LD      A,8             ;File Not Fonnd
146  0076' 37                     SCF
147  0077' C0                     RET     NZ
148  0078' E5                     PUSH    HL
149  0079' ED 5B 74 1F            LD      DE,(_IBFAD)
150  007D' 01 20 00               LD      BC,20H
151  0080' ED B0                  LDIR
152  0082' E1                     POP     HL
153  0083' 7E                     LD      A,(HL)
154  0084' CD ** **               CALL    FMCHK
155  0087' C9                     RET
156  0088'
157  0088' FE 41        DEVCHK: CP      'A'
158  008A' 38 04                  JR      C,DEVCH1
159  008C' FE 45                  CP      'D'+1
160  008E' 3F                     CCF
161  008F' D0                     RET     NC
162  0090'
163  0090' 3E 03        DEVCH1: LD      A,3     ;Bad File descripter
164  0092' C9                     RET
165  0093'
166  0093'              ;
167  0093'              ; FCB SEARCH
168  0093'              ;
169  0093' 0E 10        FCBSCH::LD      C,16            ;Directory Lengh
170  0095' ED 5B 60 1F            LD      DE,(_DIRPS)     ;Directory start
171  0099' ED 53 ** **  FCBSC1: LD      (DEBUF),DE
172  009D' 2A 64 1F               LD      HL,(_DTBUF)
173  00A0' 3E 01                  LD      A,1
174  00A2' CD 00 20               CALL    _DRDSB
175  00A5' D8                     RET     C
176  00A6' 06 08                  LD      B,8
177  00A8' 22 ** **     FCBSC2: LD      (HLBUF),HL
178  00AB' 7E                     LD      A,(HL)
179  00AC' FE FF                  CP      0FFH
180  00AE' 28 1A                  JR      Z,FCBSC4
181  00B0' B7                     OR      A
182  00B1' 28 0B                  JR      Z,FCBSC3
183  00B3' D5                     PUSH    DE
184  00B4' ED 5B 74 1F            LD      DE,(_IBFAD)
185  00B8' CD ** **               CALL    FCOMP
186  00BB' D1                     POP     DE
187  00BC' 28 0D                  JR      Z,FCBSC5
188  00BE' D5           FCBSC3: PUSH    DE
189  00BF' 11 20 00               LD      DE,32
190  00C2' 19                     ADD     HL,DE
191  00C3' D1                     POP     DE
192  00C4' 10 E2                  DJNZ    FCBSC2
193  00C6' 13                     INC     DE
194  00C7' 0D                     DEC     C
195  00C8' 20 CF                  JR      NZ,FCBSC1
196  00CA'
197  00CA' 3E           FCBSC4: DB      3EH     ; Z = 0
198  00CB' AF           FCBSC5: XOR     A       ; Z = 1
199  00CC' B7                     OR      A
200  00CD' C9                     RET
201  00CE'
202  00CE'
203  00CE'              ; File Name Compare
204  00CE' C5           FCOMP:  PUSH    BC
205  00CF' E5                     PUSH    HL
206  00D0' 06 10                  LD      B,16    ;Directory lengh
207  00D2' 13           FCOMP1: INC     DE
208  00D3' 23                     INC     HL
209  00D4' 1A                     LD      A,(DE)
210  00D5' BE                     CP      (HL)
211  00D6' 20 02                  JR      NZ,FCOMP2
212  00D8' 10 F8                  DJNZ    FCOMP1
213  00DA' E1           FCOMP2: POP     HL
214  00DB' C1                     POP     BC
215  00DC' C9                     RET
216  00DD'
217  00DD'              ; FILE MODE CHECK
218  00DD' E5           FMCHK:  PUSH    HL
219  00DE' E6 87                  AND     87H             ;1000&0111B
220  00E0' 21 1F 29               LD      HL,291FH        ;%FTYPE
221  00E3' BE                     CP      (HL)
222  00E4' E1                     POP     HL
223  00E5' C8                     RET     Z
224  00E6' 3E 06                  LD      A,6             ;Bad File Mode
225  00E8' 37                     SCF
226  00E9' C9                     RET
227  00EA'
228  00EA'
229  00EA'              ;=====================
230  00EA'              ; FILE OPEN FOR WRITE
231  00EA'              ;=====================
232  00EA' CD A3 1F     WROPEN::CALL    _FILE
233  00ED' D8                     RET     C
234  00EE' CD AF 1F               CALL    _WOPEN
235  00F1' D8                     RET     C
236  00F2'                         ;
237  00F2' 01 20 00               LD      BC,20H
238  00F5' 11 ** **               LD      DE,FILE_BF
239  00F8' 2A 74 1F               LD      HL,(_IBFAD)
240  00FB' ED B0                  LDIR
241  00FD'
242  00FD' 3A 5D 1F               LD      A,(_DSK)
243  0100' 32 ** **               LD      (FLDSK),A
244  0103'
245  0103' 2A E1 27               LD      HL,(27E1H)
246  0106' 22 ** **               LD      (HLBUF),HL
247  0109' 2A DF 27               LD      HL,(27DFH)
248  010C' 22 ** **               LD      (DEBUF),HL
249  010F'
250  010F'                        ;CALL   RDFAT           ;FAT has already loaded in _WOPEN
251  010F'
252  010F' CD ** **               CALL    FCGET
253  0112' D8                     RET     C
254  0113'
255  0113' 32 ** **               LD      (FSTCLST),A
256  0116' 32 ** **               LD      (TBLCLST),A
257  0119' 3E 80                  LD      A,80H
258  011B' 32 ** **               LD      (TBLCLST+1),A
259  011E'
260  011E' AF                     XOR     A
261  011F' 32 ** **               LD      (RC),A
262  0122' 32 ** **               LD      (LST_DSK),A     ;Never beibg then same
263  0125' 32 ** **               LD      (FLPNT),A
264  0128'
265  0128' 21 00 00               LD      HL,0
266  012B' 22 ** **               LD      (FLSIZE),HL
267  012E'
268  012E' 01 37 00               LD      BC,IMF_SIZE
269  0131' 11 ** **               LD      DE,wkout
270  0134' 21 ** **               LD      HL,FILE_BF
271  0137' ED B0                  LDIR
272  0139'
273  0139' 3E 00                  LD      A,0
274  013B' 32 ** **               LD      (WRPNT),A
275  013E'
276  013E' B7                     OR      A               ;RCF
277  013F' C9                     RET
278  0140'
279  0140'              ;==============
280  0140'              ; FILE CLOSE
281  0140'              ;==============
282  0140' 21 ** **     CLOSE:: LD      HL,wkout
283  0143' 11 ** **               LD      DE,FILE_BF
284  0146' 01 37 00               LD      BC,IMF_SIZE
285  0149' ED B0                  LDIR
286  014B'
287  014B' 3A ** **               LD      A,(FLDSK)
288  014E' 32 5D 1F               LD      (_DSK),A
289  0151'
290  0151' 2A ** **               LD      HL,(FLSIZE)
291  0154' 2C                     INC     L
292  0155' 2D                     DEC     L
293  0156' C4 ** **               CALL    NZ,WRITE1
294  0159'
295  0159' 3A ** **               LD      A,(RC)
296  015C' 2A ** **               LD      HL,(FLSIZE)
297  015F' 2C                     INC     L
298  0160' 2D                     DEC     L
299  0161' 20 01                  JR      NZ,COL3
300  0163' 3C                     INC     A
301  0164' 67           COL3:   LD      H,A
302  0165' 22 ** **               LD      (FLSIZE),HL
303  0168'
304  0168' 3E 01                  LD      A,1
305  016A' ED 5B ** **            LD      DE,(DEBUF)
306  016E' 2A 64 1F               LD      HL,(_DTBUF)
307  0171' CD 00 20               CALL    _DRDSB
308  0174' D8                     RET     C
309  0175'
310  0175' 21 ** **               LD      HL,FILE_BF
311  0178' ED 5B ** **            LD      DE,(HLBUF)
312  017C' 01 20 00               LD      BC,20H
313  017F' ED B0                  LDIR
314  0181'
315  0181' 3E 01                  LD      A,1
316  0183' ED 5B ** **            LD      DE,(DEBUF)
317  0187' 2A 64 1F               LD      HL,(_DTBUF)
318  018A' CD 03 20               CALL    _DWTSB
319  018D' D8                     RET     C
320  018E'
321  018E' CD ** **               CALL    RDFAT
322  0191' D8                     RET     C
323  0192'
324  0192' 06 10                  LD      B,10H
325  0194' 21 ** **               LD      HL,TBLCLST
326  0197' 7E           COL1:   LD      A,(HL)
327  0198' FE 7F                  CP      7FH
328  019A' 30 12                  JR      NC,COL2
329  019C'
330  019C' 23                     INC     HL
331  019D' 4E                     LD      C,(HL)
332  019E' E5                     PUSH    HL
333  019F' 2A 62 1F               LD      HL,(_FATBF)
334  01A2' 16 00                  LD      D,0
335  01A4' 5F                     LD      E,A
336  01A5' 19                     ADD     HL,DE
337  01A6' 71                     LD      (HL),C
338  01A7' E1                     POP     HL
339  01A8'
340  01A8' 05                     DEC     B
341  01A9' CA ** **               JP      Z,RDOPN3        ;Bad File Allocation
342  01AC' 18 E9                  JR      COL1
343  01AE'
344  01AE' CD ** **     COL2:   CALL    WRFAT
345  01B1' C9                     RET
346  01B2'
347  01B2'              ;=================
348  01B2'              ; INPUT FROM FILE
349  01B2'              ;=================
350  01B2' 21 ** **     INPUT_::LD      HL,RDPNT
351  01B5' 34                     INC     (HL)
352  01B6' 20 08                  JR      NZ,INP1         ;INC (RDPNT) lower byte
353  01B8'
354  01B8' CD ** **               CALL    READ
355  01BB' D8                     RET     C
356  01BC' 21 ** **               LD      HL,wkin + 31H   ;HL points FLPNT
357  01BF' 34                     INC     (HL)
358  01C0'
359  01C0' 2A ** **     INP1:   LD      HL,(RDPNT)
360  01C3' 7E                     LD      A,(HL)
361  01C4' B7                     OR      A               ;RCF
362  01C5' C9                     RET
363  01C6'
364  01C6'
365  01C6'              ;=====================
366  01C6'              ; Seek for input file
367  01C6'              ;=====================
368  01C6' 7D           FSEEK:: LD      A,L
369  01C7' 3D                     DEC     A
370  01C8' 32 ** **               LD      (RDPNT),A
371  01CB' 7C                     LD      A,H
372  01CC' 32 ** **               LD      (wkin + 31H),A  ;LD (FLPNT),A
373  01CF' CD ** **               CALL    READ
374  01D2' 21 ** **               LD      HL,wkin + 31H
375  01D5' 34                     INC     (HL)
376  01D6' B7                     OR      A
377  01D7' C9                     RET
378  01D8'
379  01D8' 00 B0        RDPNT:: DW      RDBUF
380  01DA'
381  01DA'
382  01DA' 01 37 00     READ:   LD      BC,IMF_SIZE
383  01DD' 11 ** **               LD      DE,FILE_BF
384  01E0' 21 ** **               LD      HL,wkin
385  01E3' ED B0                  LDIR
386  01E5'
387  01E5' 3A ** **               LD      A,(FLDSK)
388  01E8' 32 5D 1F               LD      (_DSK),A
389  01EB'
390  01EB' 3A ** **               LD      A,(FLPNT)
391  01EE' 47                     LD      B,A
392  01EF' 3A ** **               LD      A,(RC)
```


▶名古屋周辺では，最近“塾ってとってもヘルシー”などという妙なCMをやっています。某明○ゼミナール……。
安保　和幸（16）愛知県

```
393  01F2' B8                   CP      B
394  01F3' DA ** **             JP      C,RDOPN3        ;Bad Allocation File
395  01F6'
396  01F6' 78                   LD      A,B
397  01F7' CD ** **             CALL    PNTREC
398  01FA' EB                   EX      DE,HL           ; DE = RECORD
399  01FB'
400  01FB' 3E 01                LD      A,1
401  01FD' 21 00 B0             LD      HL,RDBUF
402  0200' CD 00 20             CALL    _DRDSB
403  0203' D8                   RET     C
404  0204'
405  0204' 01 37 00             LD      BC,IMF_SIZE
406  0207' 11 ** **             LD      DE,wkin
407  020A' 21 ** **             LD      HL,FILE_BF
408  020D' ED B0                LDIR
409  020F' B7                   OR      A               ;RCF
410  0210' C9                   RET
411  0211'
412  0211'              ;===============
413  0211'              ; PRINT TO FILE
414  0211'              ;===============
415  0211' 2A ** **     PRINT_::LD      HL,(wkout+12H)
416  0214' 23                   INC     HL
417  0215' 22 ** **             LD      (wkout+12H),HL  ;(FLSIZE)++
418  0218'
419  0218' 2A ** **             LD      HL,(WRPNT)
420  021B' 77                   LD      (HL),A
421  021C'
422  021C' 21 ** **             LD      HL,WRPNT
423  021F' 34                   INC     (HL)            ;INC (WRPNT1) lower byte
424  0220' 37                   SCF
425  0221' 3F                   CCF                     ;CY = 0
426  0222' C0                   RET     NZ
427  0223'
428  0223' CD ** **             CALL    WRITE
429  0226' D8                   RET     C
430  0227'
431  0227' 21 ** **             LD      HL,wkout+31H    ;HL points FLPNT
432  022A' 34                   INC     (HL)
433  022B' B7                   OR      A               ;RCF
434  022C' C9                   RET
435  022D'
436  022D' 00 B1        WRPNT::DW       WRBUF
437  022F'
438  022F'
439  022F' 01 37 00     WRITE:  LD      BC,IMF_SIZE
440  0232' 11 ** **             LD      DE,FILE_BF
441  0235' 21 ** **             LD      HL,wkout
442  0238' ED B0                LDIR
443  023A'
444  023A' 3A ** **             LD      A,(FLDSK)
445  023D' 32 5D 1F             LD      (_DSK),A
446  0240'
447  0240' 3A ** **     WRITE1: LD      A,(FLPNT)
448  0243' 47                   LD      B,A
449  0244' 3A ** **             LD      A,(RC)
450  0247' B8                   CP      B
451  0248' 30 76                JR      NC,WRITE4
452  024A'
453  024A' CD ** **             CALL    RDFAT
454  024D' D8                   RET     C
455  024E'
456  024E' 3A ** **             LD      A,(FLPNT)
457  0251' E6 F0                AND     0F0H
458  0253' 47                   LD      B,A
459  0254' 3A ** **             LD      A,(RC)
460  0257' E6 F0                AND     0F0H
461  0259' B8                   CP      B
462  025A' 30 47                JR      NC,WRITE2
463  025C'
464  025C' CB 3F                SRL     A
465  025E' CB 3F                SRL     A
466  0260' CB 3F                SRL     A
467  0262' CB 3F                SRL     A               ;A = A / 16
468  0264' 21 ** **             LD      HL,TBLCLST+1
469  0267' 16 00                LD      D,0
470  0269' 5F                   LD      E,A
471  026A' 19                   ADD     HL,DE
472  026B' E5                   PUSH    HL
473  026C'
474  026C' 3A ** **             LD      A,(RC)
475  026F' E6 F0                AND     0F0H
476  0271' CD ** **             CALL    PNTREC
477  0274' CD ** **             CALL    RECCL
478  0277'
479  0277' 2A 62 1F             LD      HL,(_FATBF)
480  027A' 16 00                LD      D,0
481  027C' 5F                   LD      E,A             ;A = RC && 0F0H
482  027D' 19                   ADD     HL,DE
483  027E' 36 8F                LD      (HL),8FH        ;DUMMY
484  0280' CD ** **             CALL    FCGET
485  0283' 77                   LD      (HL),A
486  0284' E1                   POP     HL
487  0285' D8                   RET     C
488  0286' 77                   LD      (HL),A
489  0287' 23                   INC     HL
490  0288' 36 80                LD      (HL),80H
491  028A'
492  028A' 2A 62 1F             LD      HL,(_FATBF)
493  028D' 16 00                LD      D,0
494  028F' 5F                   LD      E,A
495  0290' 19                   ADD     HL,DE
496  0291' 36 80                LD      (HL),80H
497  0293' 3A ** **             LD      A,(RC)
498  0296' E6 F0                AND     0F0H
499  0298' C6 10                ADD     A,16            ;ONE CLUSTER ADDED
500  029A' 32 ** **             LD      (RC),A
501  029D'
502  029D' CD ** **             CALL    WRFAT
503  02A0' D8                   RET     C
504  02A1' 18 9D                JR      WRITE1
505  02A3'
506  02A3' 3A ** **     WRITE2: LD      A,(FLPNT)
507  02A6' 32 ** **             LD      (RC),A
508  02A9' CB 3F                SRL     A
509  02AB' CB 3F                SRL     A
510  02AD' CB 3F                SRL     A
511  02AF' CB 3F                SRL     A               ;A = A / 16
512  02B1' 21 ** **             LD      HL,TBLCLST+1
513  02B4' 16 00                LD      D,0
514  02B6' 5F                   LD      E,A
515  02B7' 19                   ADD     HL,DE
516  02B8'
517  02B8' 3A ** **             LD      A,(FLPNT)
518  02BB' E6 0F                AND     0FH
519  02BD' C6 80                ADD     A,80H
520  02BF' 77                   LD      (HL),A
521  02C0'
522  02C0' 3A ** **     WRITE4: LD      A,(FLPNT)
523  02C3' CD ** **             CALL    PNTREC
524  02C6' EB                   EX      DE,HL           ;DE = RECORD NO.
525  02C7'
526  02C7' 3E 01                LD      A,1
527  02C9' 21 00 B1             LD      HL,WRBUF
528  02CC' CD 03 20             CALL    _DWTSB
529  02CF' D8                   RET     C
530  02D0'
531  02D0' 01 37 00             LD      BC,IMF_SIZE
532  02D3' 11 ** **             LD      DE,wkout
533  02D6' 21 ** **             LD      HL,FILE_BF
534  02D9' ED B0                LDIR
535  02DB' B7                   OR      A               ;RCF
536  02DC' C9                   RET
537  02DD'
538  02DD'
539  02DD'              ; FILE POINTER (A) => RECORD NO. (HL)
540  02DD' F5           PNTREC: PUSH    AF
541  02DE' F5                   PUSH    AF
542  02DF' CB 3F                SRL     A
543  02E1' CB 3F                SRL     A
544  02E3' CB 3F                SRL     A
545  02E5' CB 3F                SRL     A               ;A = A / 16
546  02E7' 21 ** **             LD      HL,TBLCLST
547  02EA' 16 00                LD      D,0
548  02EC' 5F                   LD      E,A
549  02ED' 19                   ADD     HL,DE
550  02EE' 7E                   LD      A,(HL)
551  02EF' CD ** **             CALL    CLREC
552  02F2' F1                   POP     AF
553  02F3' E6 0F                AND     0FH
554  02F5' 85                   ADD     A,L
555  02F6' 6F                   LD      L,A
556  02F7' F1                   POP     AF
557  02F8' C9                   RET
558  02F9'
559  02F9'
560  02F9'              ; FAT READ TO BUFFER
561  02F9' D5           RDFAT:  PUSH    DE
562  02FA' E5                   PUSH    HL
563  02FB' 3A ** **             LD      A,(LST_DSK)
564  02FE' 57                   LD      D,A
565  02FF' 3A 5D 1F             LD      A,(_DSK)
566  0302' BA                   CP      D
567  0303' 28 0F                JR      Z,RDFAT_1
568  0305' 32 ** **             LD      (LST_DSK),A
569  0308' 3E 01                LD      A,1
570  030A' ED 5B 5E 1F          LD      DE,(_FATPOS)
571  030E' 2A 62 1F             LD      HL,(_FATBF)
572  0311' CD 00 20             CALL    _DRDSB
573  0314' E1           RDFAT_1:POP     HL
574  0315' D1                   POP     DE
575  0316' C9                   RET
576  0317'
577  0317'              ; FAT WRITE FROM BUFFER
578  0317' D5           WRFAT:  PUSH    DE
579  0318' E5                   PUSH    HL
580  0319' 3E 01                LD      A,1
581  031B' ED 5B 5E 1F          LD      DE,(_FATPOS)
582  031F' 2A 62 1F             LD      HL,(_FATBF)
583  0322' CD 03 20             CALL    _DWTSB
584  0325' E1                   POP     HL
585  0326' D1                   POP     DE
586  0327' C9                   RET
587  0328'
588  0328'              ; FREE CLUSTER POSITION GET
589  0328' C5           FCGET:  PUSH    BC
590  0329' E5                   PUSH    HL
591  032A' 06 80                LD      B,80H
592  032C' 2A 62 1F             LD      HL,(_FATBF)
593  032F' 7E           FCGET2: LD      A,(HL)
594  0330' B7                   OR      A
595  0331' 28 08                JR      Z,FCGET3
596  0333' 23                   INC     HL
597  0334' 10 F9                DJNZ    FCGET2
598  0336' 3E 09                LD      A,9             ;Device Full
599  0338' 37                   SCF
600  0339' 18 04                JR      FCGET4
601  033B' 3E 80        FCGET3: LD      A,80H
602  033D' 90                   SUB     B
603  033E' A7                   AND     A
604  033F' E1           FCGET4: POP     HL
605  0340' C1                   POP     BC
606  0341' C9                   RET
607  0342'
608  0342'              ; CLUSTER (A) => RECORD (HL)
609  0342' 26 00        CLREC:  LD      H,0
610  0344' 6F                   LD      L,A
611  0345' 29                   ADD     HL,HL
612  0346' 29                   ADD     HL,HL
613  0347' 29                   ADD     HL,HL
614  0348' 29                   ADD     HL,HL
615  0349' C9                   RET
616  034A'
617  034A'              ; RECORD (HL) => CLUSTER (A)
618  034A' E5           RECCL:  PUSH    HL
619  034B' CB 3C                SRL     H
620  034D' CB 1D                RR      L       ;HL/2
621  034F' CB 3C                SRL     H
622  0351' CB 1D                RR      L       ;HL/4
623  0353' CB 3C                SRL     H
624  0355' CB 1D                RR      L       ;HL/8
625  0357' CB 3C                SRL     H
626  0359' CB 1D                RR      L       ;HL/16
627  035B' 7D                   LD      A,L
628  035C' E1                   POP     HL
629  035D' C9                   RET
630  035E'
631  035E'              ;
632  035E'              ; Works
633  035E'              ;
634  035E'
635  035E'                      DSEG
636  0000"
637  0000"              FCB_ADR:DS      2               ;File Control Block Address
638  0002"              LST_DSK:DS      1               ;Disk drived last
639  0003"
640  0003"
641  0037               IMF_SIZE        EQU     37H
642  0003"              wkin::  DS      IMF_SIZE        ;work area for Input-file
643  003A"              wkout:: DS      IMF_SIZE        ;work area for Output-file
644  0071"
645  0071"              FILE_BF:DS      12H
646  0083"              FLSIZE: DS      2   ;File Size.
647  0085"              FLDTADR:DS      2   ;Start Address.
648  0087"              FLEXADR:DS      2   ;Exec Address.
649  0089"                      DS      6
650  008F"              FSTCLST:DS      1   ;First Cluster.
651  0090"                      DS      1
652  0091"              TBLCLST:DS      10H ;Cluster table.
653  00A1"              FLDSK:  DS      1   ;The Login disk.
654  00A2"              FLPNT:  DS      1   ;The FILE Pointor.
655  00A3"              DEBUF:  DS      2   ;Record No. Which Have The DIR.
656  00A5"              HLBUF:  DS      2   ;Address Where On IMFORMATION.
657  00A7"              RC:     DS      1   ;The Number of Records The File have.
658  00A8"
659  00A8"                      END
```

リスト4 SOS DEF

```
 1 _HOT     EQU     1FFAH
 2 _PRINT   EQU     1FF4H
 3 _PRNTS   EQU     1FF1H
 4 _LTNL    EQU     1FEEH
 5 _NL      EQU     1FEBH
 6 _MSX     EQU     1FE5H
 7 _TAB     EQU     1FDFH
 8 _GETL    EQU     1FD3H
 9 _PAUSE   EQU     1FC7H
10 _PRTHX   EQU     1FC1H
11 _PRTHL   EQU     1FBEH
12 _HEX     EQU     1FB8H
13 _FILE    EQU     1FA3H
```


▶本屋でバイトをしている私。先日お客さんに「角川文庫でハンムラビ王の書いた“太陽〜”はありますか」ときかれ，一瞬考えこみました。ハンムラビ王って本なんてだしてたっけ……？　なんのことはない，「半村良」を聞き間違えたのでした。私も老いたな。
中村　健（20）埼玉県

```
14 _WOPEN  EQU      1FAFH
15 _POKE   EQU      1F9AH
16 _PEEK   EQU      1F94H
17 _ROPEN  EQU      2009H
18 _KILL   EQU      2015H
19 _NAME   EQU      2012H
20 _ERROR  EQU      2033H
21
22 _DRDSB  EQU      2000H
23 _DWTSB  EQU      2003H
24
25 _PRCNT  EQU      1F7AH
26 _KBFAD  EQU      1F76H
27 _IBFAD  EQU      1F74H
28 _SIZE   EQU      1F72H
29 _MEMAX  EQU      1F6AH
30 _DTBUF  EQU      1F64H
31 _FATBF  EQU      1F62H
32 _DIRPS  EQU      1F60H
33 _FATPOS EQU      1F5EH
34 _DSK    EQU      1F5DH
35
```

リスト5 WLB.DEF

```
 1 ;============================
 2 ; WLB.DEF Header File For WLB
 3 ;      # WZD
 4 ;      #=WLB1
 5 ;      *=WLB2
 6 ;      # WLK /P:3000,/D:4500
 7 ;      * WLB1,WLB2,WLB/N:P
 8 ;      #
 9 ;      CSEG      3000H
10 ;      DSEG      4500H
11 ;
12 ;============================
13
14 RDBUF   EQU      0B000H  ;- B0FFH
15 WRBUF   EQU      0B100H  ;- B1FFH
16
17 cmdbuf  EQU      5000H
18
```

全機種共通システムインデックス

■85年 6 月号
序論　共通化の試み
第 1 部　S-OS"MACE"
第 2 部　Lisp-85インタプリタ
第 3 部　チェックサムプログラム
■85年 7 月号
第 4 部　マシン語プログラム開発入門
第 5 部　エディタアセンブラZEDA
第 6 部　デバッグツールZAID
■85年 8 月号
第 7 部　ゲーム開発パッケージBEMS
第 8 部　ソースジェネレータZING
■85年 9 月号
インタラプト　S-OS番外地
第 9 部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年 1 月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年 2 月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年 3 月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年 4 月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年 5 月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年 6 月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD" 2000 QD
連載　対話で学ぶ magiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年 7 月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版 S-OS"SWORD"
■86年 8 月号
第26部　対局五目並べ
第27部　MZ-2500版 S-OS"SWORD"
■86年 9 月号
第28部　FuzzyBASIC 発表
連載　明日に向かって magiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタ DREAM
第31部　FuzzyBASIC 料理法〈1〉
■86年11月号
第32部　パズルゲーム HOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC 料理法〈2〉
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC 料理法〈3〉
■87年 1 月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC 料理法〈4〉
■87年 2 月号
第36部　アドベンチャーゲーム MARMALADE
第37部　テキアベ作成ツール CONTEX
■87年 3 月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツール MAGE
付録　"SWORD" 再掲載と MAGIC の標準化
■87年 4 月号
第40部　INVADER GAME
第41部　TANGERINE
■87年 5 月号
第42部　S-OS"SWORD" 変身セット
第43部　MZ-700用 "SWORD" を QD 対応に
■87年 6 月号
インタラプト　コンパイラ物語
第44部　FuzzyBASIC コンパイラ
第45部　エディタアセンブラ ZEDA-3
■87年 7 月号
第46部　STORY MASTER
■87年 8 月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージ JACKWRITE
特別付録　FM-7/77版 S-OS"SWORD"
■87年 9 月号
第49部　リロケータブル逆アセンブラ Inside-R
特別付録　PC-8001/8801 版 S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASIC コンパイラの拡張
第52部　X1turbo 版 S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OS の仲間たち
第53部　もうひとつの FuzzyBASIC 入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OS こちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo 版 "SWORD" アフターケア
　　　　ラインプリントルーチン
特別付録　PASOPIA7 版 S-OS"SWORD"
■88年 1 月号
第58部　FuzzyBASIC コンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年 2 月号
第59部　シューティングゲーム ELFES
■88年 3 月号
第60部　構造型コンパイラ言語 SLANG
■88年 4 月号
第61部　デバッギングツール TRADE
第62部　シミュレーションウォーゲーム WALRUS
■88年 5 月号
第63部　シューティングゲーム ELFES II
第64部　地底最大の作戦
■88年 6 月号
第65部　構造化言語 SLANG 入門(1)
第66部　Lisp-85 用 NAMPA シミュレーション
■88年 7 月号
第67部　マルチウィンドウドライバ MW-1
連載　構造化言語 SLANG 入門(2)
■88年 8 月号
第68部　マルチウィンドウエディタ WINER
■88年 9 月号
第69部　超小型エディタ TED-750
第70部　アフターケア WINER の拡張
■88年10月号
第71部　SLANG 用ファイル入出力ライブラリ
第72部　シューティングゲーム MANKAI
■88年11月号
第73部　シューティングゲーム ELFES Ⅳ
■88年12月号
第74部　ソースジェネレータ SOURCERY
■89年 1 月号
第75部　パズルゲーム LAST ONE
第76部　ブロックゲーム FLICK
■89年 2 月号
第77部　高速エディタアセンブラ REDA
特別付録　X1版 S-OS"SWORD"〈再掲載〉
■89年 3 月号
第78部　Z80用浮動小数点演算パッケージSOROBAN
■89年 4 月号
第79部　SLANG 用実数演算ライブラリ
■89年 5 月号
第80部　ソースジェネレータ RING
■89年 6 月号
第81部　超小型コンパイラTTC
■89年 7 月号
第82部　TTC用パズルゲーム　TICBAN
■89年 8 月号
第83部　CP/M用ファイルコンバータ
■89年 9 月号
第84部　生物進化シミュレーションBUGS
■89年10月号
第85部　小型インタプリタ言語TTI
■89年11月号
第86部　TTI用パズルゲーム PUSH BON!
■89年12月号
第87部　SLANG用リダイレクションライブラリ
　　　　DIO. LIB
■90年 1 月号
第88部　SLANG用ゲームWORM KUN
特別付録　再掲載SLANGコンパイラ
■90年 2 月号
第89部　超小型コンパイラTTC++
■90年 3 月号
第90部　超多機能アセンブラOHM-Z80
■90年 4 月号
第91部　ファジィコンピュータシミュレーションI-MY
■90年 5 月号
第92部　インタプリタ言語STACK
■90年 6 月号
第93部　リロケータブルフォーマットの取り決め
第94部　STACK用ゲーム SQUASH!
第95部　X68000対応S-OS"SWORD"
特別付録　PC-286対応S-OS"SWORD"
■90年 7 月号
第96部　リロケータブルアセンブラWZD
■90年 8 月号
第97部　リンカWLK
■90年 9 月号
第98部　BILLIARDS

＊以上のアプリケーションは，基本システムである S-OS "MACE" または S-OS"SWORD" がないと動作しませんのでご注意ください。

THE SENTINEL

▶イラクはクウェートを占領した。原油の供給量はやはり減るだろう。金が上がった，株が下がった。ところで，こんなときだから油田でも掘り当てるゲームはいかがなもんでしょう。
阿妻　靖史 (18) 東京都

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1990年10月18日の到着分までとします。当選者の発表は1990年12月号で行います。

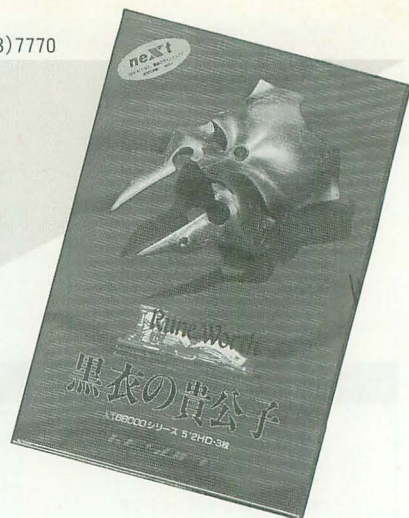
T&E SOFT ☎052(773)7770

ルーンワース ～黒衣の貴公子

X68000用 5"2HD版2枚組

8,800円

3名

プレイヤーのとる行動によって，エンディングが変わる，マルチエンディングタイプのRPG。それだけに自由度が高く，遊びやすい。

2

SPS ☎0245(45)5777

プロテニス ワールドコート

X68000用 5"2HD版2枚組

8,800円

3名

ゲームセンターで人気だったナムコのテニスゲームの移植版。友達とわいわいやるのがマル，のゲームだ。

3

システムサコム ☎03(635)7609

闇の血族

X68000用
5"2HD版3枚組

8,800円

3名

サコムのノヴェルウェアシリーズ最新作。超能力を持つ美少女魅由が活躍するアドベンチャーゲームだ。

満開製作所 ☎03(554)9282

電脳倶楽部オリジナルグッズ

a. Tシャツ(M, L)	各10名
b. 豆しぼり	10名
c. えんぴつ	10名
d. シャープペン	10名
e. ボールペン	10名

お馴染み満開製作所から，Oh! Xの読者にオリジナルグッズをプレゼント。全部で60名にさしあげます。さぁすが，太っ腹！（Tシャツはサイズ明記のこと）

8月号プレゼント当選者

1ディスプレイ（東京都）牧野博史　2漢字プリンタ(埼玉県）加藤健二　3イメージスキャナ（大阪府）追田賢一　4サイバースティック（宮城県）及川雄也　5数値演算ボード（愛知県）佐原功治　6MIDIボード（北海道）米田幸弘　7RAMボード（兵庫県）松川努　8ビデオボード（京都府）笹尾兼右　9延長ケーブル（埼玉県）桑原亜砂雄（千葉県）福富友宏（神奈川県）古木健一　信太徹　鈴木康之（愛知県）山本佳孝　安尾文教(三重県）小掠昌宏（大阪府）土岐太司（愛媛県）渡部圭誠

以上の方々が当選されました。おめでとうございます。商品は順次発送いたしますが，入荷状況などにより遅れる場合もあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますのでご了承ください。

（価格はすべて消費税別です）

◆X-OVER NIGHT

(クロスオーバーナイト)

［第5話］

2つの人種？

TAKAHARA HIDEKI 高原 秀己

先日，ある雑誌に載っていた短いコラムに次のようなことが書いてあった。

「現代の若者はバンドを組んで汗を流して大騒ぎするタイプと，部屋の中でAV（2通りの読み方があるようだがどちらに読んでいただいても結構）やパソコン，ファミコンに熱中するタイプとに分かれる。この両者は“人種”としても分かれており，後者は前者に比べて生殖能力が劣るのではないか？」

このコラム，自分はやっぱり後者だろうなあ，などと思いつつ読むとなかなかイヤなところをついてくる。

暴走少年たちやディスコ大好き少女は前者と同じグルーピングとなる。われわれのような“コンピュータ少年”たちは後者に属するわけで，いわれてみると，30歳を過ぎた独身者がゴロゴロしているのはもちろん，楽しそうに日夜デートにいそしむ若者はかなり少ないように感じる。逆に髪の毛を金色に染めた，一見してそれとわかるバンド青年たちが，右手に楽器，左手に同様の人種の女の子を抱えて楽しそうに生きている光景はよく見かける。

先のコラムには「触れ合うとすぐに妊娠してしまいそうなエネルギーを彼らは発散している」という感じで書いてあった。オーバーではあるが，的は外していない。夜に髪を金色に染めた男女のカップルが仲よく電車を降りていったりすると，その後の彼らのなりゆきを無意識に想像してしまったりする。逆にドラゴンクエストを並んで買って一目散に帰る大学生が，実は女性と同棲していて……というのはあまりにもイメージがわかない。

＊　＊　＊

なるほど，ベーシックに考えてみればみるほど，この発想に大きな誤りがないような気がしてくる。

歌と踊りとは人間の楽しみのうち，最も原始的なものであるという。スピード，暴力もしかり。シリコンチップによる産物のコンピュータなどという存在との交流とは対極を成すものである。

恋愛－セックスというプロセスも極めて基本的な本能に基づく人間の行為である。バンド少年や暴走少年たちのほうが，このことひとつをとっても有利なポジションにいる。ちなみに人間が最もその気になるのは，スピードを出し過ぎて疲労に襲われているときだ，という説もある。

しかし，そもそも，バンド少年にしろ，ディスコ少女でもサーファーでもいいのだが，こういったタイプは他人とのインタフェイスを持つことが日常茶飯事だし，仲間同士で肌を寄せ合って汗を流して共同作業をすることが多い。

同じ目標に向かって生きる男女が出会って，しかも共通体験をしながら生きている，ということで，恋愛感情が極めて生まれやすい土壌である。「触れ合えば妊娠」はともかくとして，フラフラになるまで演奏をしたり，背中に女性を乗せてバイクで飛ばしたあとに，恋愛感情が生まれるケースは十分にあるはず。

＊　＊　＊

一方，条件は極めて悪いながら，コンピュータ族にとって，同類の男女の接触はないのかといえば，そうではないのが現代社会の面白いところ。ある証券会社では，

「入社したときに手の届く範囲に結婚相手がいる」

などという定説（？）があるそうだが，さほど職場結婚，職場恋愛が急増中であるそうだ。プログラマ，SEもこれに近い状態だとか。女性プログラマが増え続けていることを考えると，そうそう悲観したものではないのだろう。

もうひとつ。ネクラなコンピュータ人種同士の究極の出会いの場，といわれることの多いパソコン通信ネット。ここでは意外なほどカップルが登場するケースは多いようで，ぼくのようにネットワークの中で生活していると，あちこちでそのような成約事例を目にする。

チャット（文字会話）でやりとりをしている間に恋愛感情が生まれ，最初に実際に出会った日にホテルに行ってしまった，という伝説めいた事例もあったとか。そこまでいかなくても，パソコン通信ネットが知り合う場となり，実際に何回か会ううちに交際が始まるというケースはかなりあるようだ。まあ，結局のところ，出会いの機会があれば，事態は開けてくるということではないだろうか。

＊　＊　＊

近年起こった2大凶悪事件といえば，女子高生コンクリート詰め殺人事件と，連続幼女誘拐殺人事件。例のコラムの分類によると，この2つの事件は双方の“人種”が起こした事件の典型であるようだ。

アニメファンやパソコンフリークは，

「自分はMとは違う！」

と顔色を変えて強調するが，どちらかに振り分ければ，やはりM側のグループに属してしまうのは否めない。明らかにそちらに所属するぼく自身もそう思う。

バランスが崩れると，何が起こるかわからないのが，現代社会の恐ろしいところである。だからどんな方法であっても，バランスを崩さないためには，**僕の横にもカワイイ女の子がいる必要がある。**

なんたって，街を歩くカップルの図はいかにもバランスがいい。逆に重そうなショルダー式のカバンをぶら下げた疲れた男とでは，見た目のバランスはどちらがいいかはいうまでもないであろうから。

第42回　知能機械概論――お茶目な計算機たち――

続・超能力実験の成果〈透視〉

視聴者のだまし方

超能力と称したトリック（あるいはインチキ）に僕たちはずいぶんと慣らされてきました。そしてその結果，多くの人は超能力現象の存在を信じているというわけでもなく，また否定するわけでもなく，ただ適度に楽しんでいるように思われます。

「どうせ，エンターテイメントさ」という開き直りは切り札として持ってはいるものの，司会者までがグルだという噂を聞くとちょっと憤りを感じてしまう程度の真面目さは，まだ持ち合わせているという人も少なくないかもしれません。僕自身も，真面目にニュースを話す人と奇術師とつるんで視聴者をだます人は違っていてほしいと思います。

そのような，いかにもありそうなごまかしのテクニックとは違い，もうひとつの悪質な嘘といえば，数学的知識，確率的な計算結果を暗に，あるいは露骨に利用して視聴者をごまかす方法です。科学的客観的な背景を利用して権威づけしてごまかすのですから，「司会者＝サクラ」とはまた別な意味であるいはもっと悪質だといえます。

確率を利用して嘘をつくやり方で一番単純な方法は，何回試したかということを明らかにしないというやり方です。

たとえば，1000回程度に1回しかうまくいかないようなこと，ここでは「コインを10枚まとめて投げ，すべて表を出す」というのを考えてみましょう。計算してみれば，まったく機械的に振った場合，このようなことが起こる確率は，1024分の1です。これは確かに，ずいぶんと珍しい事象なのですが，逆にいえば，1000回ぐらい試し続ければ1度ぐらいはうまくいくといえるのです。ですから，うまくいったときのビデオだけを編集して見せられても，なんとも判断のしようがないわけです。

この間も，朝のテレビで超能力モドキをやっていました。なんだかんだと多数のうさんくさい実験をやっているのですが，その中にこんなのがありました。

5種類のマークのついたカードを用意し，その中から1枚を引き抜き，それを裏のまま4人（そのうち3人は素人という設定）が透視して当てるというものでした。透視する順番がミソでして，向かって左の人から順番に予想をいい，もし，同じマークをすべての人がいったときはそれでいいのですが，違うマークを予想した時点で終了し，カードを表にするのです。ここがチェックなのですが，超能力者は一番右に座っているのです。ですから，超能力者はめったに予想しないのです。

当然，うまく4人の予想が一致し，表を開くとそのとおりだったという例がビデオで流されました。さて，このような例が出現する確率を考えてみましょう。ひとりが当たる確率は1/5ですから，4人いるからその4乗で，625分の1，つまり625回に1回起こるというぐらい珍しいことであるといえます。

レポーターは比較的よく見る人でして，その実験で左から2番目の人を務めていました。ここで，司会者はサクラでないと考えてみましょう。でも，500回も600回も実験をやったとはとうてい考えられません。となると考えられることはずいぶん限定されてきます。

まず，超能力者の予想の仕方があるのです。超能力者は前の3人が同じマークをいったら必ずそれと同じマークをいうのです。これにより確率はさっきの計算の5倍になり125回程度やればよいことになります。さらに，超能力者のすぐ左の人は超能力者が連れてきた人かもしれませんので，同じようにいっていると考えられます。

となると，一番左の人とその次の司会者がうまく当てさえすれば，残りの2人はいないのと同じということになり，結局確率は，たったの1/5の2乗，つまり25回ぐらいやるだけで4人全員の予想が一致し，しかもそれが裏返しのカードと当たってしまうということになります。なんというトリックなのでしょう。

この超能力実験に秘められた疑惑を列挙してみましょう。

1.失敗した例は1回ぐらいしか見せないので何回やったかわからない。

2.超能力者は毎回予想をしなくてよいので，ボロが出ない。

3.超能力者（ともうひとり）は前の人と同じことをいうので，当たる確率は大幅に増えている。

もちろん，最初っからすべて（番組制作側が）インチキという可能性だってありますけどね。確率を計算しましょうか？

透視実験の方法

先月紹介した超能力っぽい現象はそれこそ偶発的なできごとだったのですが，今月はちょっとだけ厳密に実験を行いました。そして，その結果の解析についても少し確率計算に基づいて行いました。

超能力といってもいろいろな種類があるとされているようですが，ここで対象としたのは先ほど批判したテレビ番組と同じように透視です。先月紹介したライン教授が使用したのとおそらく同じタイプのゼナーカードを使用します。図1にそのカードに描かれている5種類のパターンを示します。

実験の方法は次のとおりです。図1のような5種類のパターンが描かれているカードを1枚ずつ目印として表を向けて横1列に並べます。そして残りの25枚（各パターン5枚ずつ）を裏返しにしてよく切っておきます。

透視実験は裏返しにしたそのカードのパターンを1枚ずつ予想することによって行います。これだなと思ったパターンのところへ裏返しのまま（下方向に並べて）置いていきます。

これを25枚すべてについて行うと1回の試行が終了というわけです。それから結果を調べます。すべてのカードを表にして，予想（透視）どおりのカード（最初に表を向けたカードのパターンと一致しているカード）の枚数を数えます。この枚数は最小なら0で最大ならばすべて当たったときで25ということになります。

注意するべきことは1枚透視するたびに表にして結果を見ないことと，各パターン5枚ずつというようにわざと枚数を合わせる必要はないということです。前者は，いままでの結果から次のカードの予測のためのデータを得ないようにするためであり，後者は，最後のほうで枚数合わせのために透視をしなくなってしまうことを避けるためです。

透視実験だけでなく，2人で行うテレパシー実験も記録しようと作ったのですが，今回は透視実験だけを行いました。ひとりに対して試行を10回やってもらいます。1回当たり25枚のカードの透視実験を行うのですから，1人当たり250回透視することに

なります。

透視実験の結果

実験結果を表1に示します。10人（Aさんから Jさん）についてのデータを当たった枚数の小さい人から載せています。結局250回の10人分ということで，2500回の試行を行ったわけです。表の横1行はある1人についてのデータであり，10回の試行それぞれの結果，そして最後に合計が書かれています。裏側を向いた1枚のカードの種類が当たる確率は，まったくランダムに予想したとすれば，5回に1回当たる勘定ですから，確率1/5，したがって10回の試行の合計では，250×1/5ですから50になります。

実験結果を見ればわかるように50を上回っているのは6人，下回っているのは3人で，全体に成績はいい（透視の実在を裏づける方向にあるというと誤解を招くか？）ようです。一番成績のよい値は1回の試行で見てみると，Jさんの13ということになります。13というと半分以上ですからかなりのものです。合計成績でもJさんの63回が最高です。

このようにただ結果を出したのではなんだかピンときませんので，ちょっと確率を使って定量的に調べてみることにしましょう。表2に1回の試行で当たる回数ごとの確率を示します。計算の方法は簡単な算数で，式で書くなら，$(1/5)^r \times (4/5)^{25-r} \times {}_{25}C_r$となります。ここでrは当たった回数，${}_{25}C_r$は25からr取り出す組み合わせです。

なお，表のカッコの中は，別の表現方法で書いたものであり，たとえば，一番起きる可能性の高い5回のところのカッコの中は，1.96……かける10の－1乗のことでして，結局左に書いてある数字と同じということになります。有効数字を考慮するときや，だいたいの大きさを素早く把握するときなどには便利です。

実験結果の中の最大値である13回当たる確率はというと，10000分の2.92……ということですから3400回ぐらいやれば，1回出る程度の確率ということになります。13回以上当たる確率は10000分の3.69……，つまり2700回に1回ということになります。今回の場合，総試行回数2500回で1回出たのですから，たいしたことないともいえます。

ちなみに1枚も当たらない確率は0.00377……ということですから，これもかなりな数だといえます。確率的には11枚以上当たる確率と同じぐらいになります。

今度は1人当たりの総試行回数250回単位で捉えてみることにします。表3に250回の試行で当たる回数とその確率をあり得そうな回数についてだけ載せておきます。今度の表には，その回数だけ当たる確率だけでなく，その回数以上当たる確率も同時に示しています。

たとえば，70回ちょうど当たる確率は，左側の数字0.000570……であり，70回以上当たる確率は右側の数字0.00148……ということです。ですから，もし自分でこの実験を行い250回の試行をやられた読者の方が1000人おられたとしたならば，1人から2人ぐらいは，この表に載っていないような大きい回数になって歯がゆい思いをするというわけなのです。

ちなみに，回数を重ねてもコンスタントにこの表をはみ出すような方が万が一いらっしゃるのならば，これははっきりいって超能力であるということができます。飛んで行きますのでご連絡ください。

25回中で13回以上当たる確率は0.000369……でした。もし，このペースで当たり続けるとしたら，250回で130回以上ということになります。このような確率は，この表には載っていませんが，計算してみると実は0が29回も続くぐらいの試行を繰り返さないと起こりそうにもないようなあり得ない数になってしまいます。

超能力者と判断するのは誰？

今回の実験結果を見て，結局何が言えるのでしょうか？　もちろん知りたいのは超能力の存在に関する確証です。ですから，できるだけ客観的に解析しようとしたのですが，どうも中途半端になりそうで，尻切れにしてしまったような気がします。

どうしてかというと，いくら確率や統計的な計算をして客観点なデータを出そうとしても，その数字をどう解釈するかということは，結局はそのデータを得た個人個人に委ねられるのだと思ったからです。

たとえば，Jさんは13枚も当たるという，10000回に3.7回ぐらいしか起こらないような枚数を当て，10回の試行を足しても，63回という100回に2.6回程度しか出ないような結果を1回やっただけで出したのだ，といっても，あまりピンとはこないような気がします。データをもっと並べてもよけいわからなくなるでしょう。

このようなことは全然別の例を出したほうがわかりやすいかもしれません。将来，天気の予報が完璧なものになって，明日，雨が降る確率が11.28%とわかったとしましょう。しかし，だからといって，この数字をどう受け取るかは人によって異なる，というよりはむしろ，責任は受け取る側にあると思います。もちろん，予想が完璧になれば，雨か晴れか100%確実にわかるのだと主張する人もいるかもしれませんが，それは理論的に不可能ではないかと思われます（素粒子の振る舞いが完全に確定できるのならば話は別でしょうが）。

科学はあくまで厳密に客観的に真理というものを追求してきたように見えます。ただ別の面では，逆に主観的な最終的な判断や責任とでもいえるものを逆にだんだんと人間のほうにつきつけてくるようになってきたといえるのではないでしょうか。

表1　実験結果

	1	2	3	4	5	6	7	8	9	10	合計
A	4	5	6	2	4	9	3	3	6	1	43
B	4	4	2	4	6	2	7	4	5	5	43
C	3	3	7	3	5	8	7	5	1	6	48
D	4	4	3	7	6	5	7	4	4	6	50
E	6	1	4	3	5	4	10	7	7	5	52
F	6	3	8	5	3	6	3	2	8	9	53
G	2	7	6	6	7	2	5	6	7	5	53
H	7	5	6	11	5	5	4	2	9	7	61
I	8	5	4	6	5	6	7	5	9	6	61
J	4	7	7	2	5	3	13	7	5	10	63

図1　ゼナーカードの5種類のパターン

表2　1回の試行の確率分布

当たる枚数	確率
0	0.00377789318629572150(3.777893e-03)
1	0.02361183241434825900(2.361183e-02)
:	:
24	0.00000000000000033554(3.355443e-16)
25	0.00000000000000000336(3.355443e-18)

表3　10回の試行の確率分布

当たる回数n	n回当たる確率	n回以上当たる確率
37	0.0073037526438598562(7.303753e-03)	0.98610628083249129000(9.861063e-01)
38	0.01023486390225098600(1.023486e-02)	0.97880252818863144000(9.788025e-01)
:	:	:
69	0.00088305295158387890(8.830530e-04)	0.00236958710962642620(2.369587e-03)
70	0.00057083065798815050(5.708307e-04)	0.00148653415804254720(1.486534e-03)

第52回

猫とコンピュータ

「文豪」が飛んだ

Takazawa Kyoko
高沢 恭子

狭い部屋をいかにうまく使うか，これは東京に住むうえでの重要なポイント。キョウコさんちでもちょっとした“改造”が行われたのですが，やはりホンニャアがいるとなかなかコトはうまく運ばないようで……。

きのうの熱が部屋の中にまだ漂っているというのに，新しい太陽は，また昇ってくる。風をもとめて開いた窓も，じっと息をひそめているだけ。

早く目ざめたつもりでも，リビングはもう光がいっぱいで，休日の朝はとっくに始まっていたらしい。食卓の上の読みかけの本だけが，ゆうべのページをひろげたまま，時を止めている。

そんな朝でも，マシンルームをのぞくと独特のひんやりした静けさがあって心地よい。きょうは日曜日だから，夫と2人，この部屋にちょっとした改造をくわだててみた。

毎日，変わらぬパワーで攻めてきたお日さまも，わずかずつ優しさをみせて，夏は遠ざかろうとしている。夏休みのリズムが身についたと思うまもなく，トオルには新学期が待っている。

うしろがコワイ

こどものころ新宿の家にいた白猫のチロは，すぐに誰かの肩の上に飛び乗ってきて困った。

困ったといっても，それは子猫のときに肩に乗せて遊んだことが習慣になったためで，原因をこしらえたのは，私だった。

小さなうちはよかったのだけれど，チロはオトナになってもこれをやめなかったから，まったく困った。しかも相手は私ばかりでなく，誰彼かまわずやった。

はじめに肩に乗せたとき，子猫も前向きにしていたので，チロは自分から乗ってくるときも，かならず人間の背中がわから飛び乗るようになった。つまり，チロが肩に乗ろうとしているとき，人間のほうは気づかないことが多いわけだ。

子猫のころは毛糸玉のように軽かったチロだが，オトナになって3kgを越えた体になると，そっと乗せてもズシリとこたえる。それが，何も知らないでいる後ろ向きの肩をめがけて，1，2mも離れたところから，跳躍力にまかせて飛びついてくるのだ。ねらわれたほうの驚きと衝撃はいつもたいへんだった。

あるとき，ガレージの隅にある整理棚の前で，何かしごとをしながら背を向けて立っていた父に，ブルーバードの屋根に乗っていたチロは，絶好の期待をかけて突進した。こともあろうに，その瞬間振り向いた父の顔と，飛びつくためにすでにツメ出していたチロは，凄絶な空中衝突をとげたのだ。

動物好きだった父はチロに憤慨することはなかったけれど，チロのほうは深い反省があったのか，それ以後，父の肩はねらわないようになった。

同じように青い目をした白猫，わが家のホンニャアが，なぜ夫の肩にだけ飛び乗るようになったのかわからない。

あるいは，夫はじぶんでも気づかずに，何回か肩に乗せたことがあって，ホンニャアもそれを交流のひとつに取り入れるようになったのかもしれない。

とくに朝の洗顔中に夫が体を前にかがめるとき，かならずといっていいほどジャンプしてくる。傾斜が真横に近いときは，背中のまんなかでしばらくくつろぎ，体が起きると，肩に移動して，夫のこめかみのあたりを熱心にナメあげる。ザラザラの舌は耐えかねるような痛さだそうだ。

ホンニャアの場合，相手は夫に限られていて，スペースの足りなそうな私や，安心感にとぼしいトオルは対象にされない。

そのジャンプが，きょうはマシンルームで行われた。

矛盾の出入り口

マシンルームは6畳ほどの大きさだ。

転勤で住まいが変わるたび，生活全体の見直しができるというのはとても得がたいことなのに，面積だけを見て，「東京に近づくにつれて，家が小さくなるネ」と，新宿のおばあちゃんは言う。

弟夫婦と3人暮らしで，ジャンボな部屋をいくつも含めた7LDKもの家に住む母から見たら，わが家はエレベータみたいに小さいそうだ。

でも，必要部分だけでできているコンパクトな家に住みなれてしまうと，この便利さからは離れがたいものがある。メンテナンスや冷暖房の簡略さも経済的だ。それにどうしても，広さに代わる機能面でのくふうに力を入れるようになるから，この楽しみは，またこたえられない。

広い家に，無用なものを並べて死んだ空間をいくつもつくるより，無限に近い機能を持った最小限のコーナーがほしい。マシンルームはそんな希望をかなえる，たのもしい部屋なのだ。

ここはわが家が目ざしているマルチ空間として，少しずつ成長をつづけてきた。15年にわたるパソコンの旅から，それなりのくふうも生まれて，むしろ以前よりシンプルなシステム配置で，豊富な機能が発揮できるようになってきた。

一見，簡素に見えるこの部屋は，望みしだいで書斎になり，図書館になり，ゲームセンターになる。いくつもの使いみちがあるとき，その場所は小さいほど役割が大きく評価される。

マシンのおかげで，データや文書は，もっとも小さく圧縮して保存でき，必要ならいつでも引き出してみることができる。だ

から，マシンと周辺機器，それにイスがひとつあるだけで，書類も原稿もディスクの中に吸い込まれてしまう。

だが，じつのところマシンルームも矛盾を抱えている。情報の圧縮をしてくれる便利なはずのマシンが，わが家には正反対の結果をもたらす一面を持っているのだ。ひとつはパソコン関係の書籍の増加と，もうひとつはパソコン通信だ。

通信は商業データベースと，草の根通信(BBS)の両方を利用している。夫はビジネス関係でNIFTY-serveや，化学技術関係専門のネットSTNへのアクセスが多く，私はもっぱら若いお友だちにあうために，FBI-NETに通いつづける。

データベースで検索した記事や情報は，プリントアウトして書類化したほうが扱いやすい。くりかえして確認もでき，複写の必要にも応じられる。これは資料としての必然性があるが，問題なのはFBIやナツメなどのBBS通信だ。

5年前の創始時期のような熱中とはちがうものの，アクセスするたび，活気のある親密なメッセージが尽きることなく書き込まれている。連載の感想も聞かせてくれるし，イベントの打ち合わせもある，電報も飛んでくる。そうなるとダウンロードしないなんて考えられない。

ダウンロードとは，通信内容をわが家の記録装置におさめることだ。これを編集してディスク内におさめておけば問題はないのだが，どうしてもプリントアウトして，内容を活字で確認したくなる。それも家族がひととおり目を通してしまえば，その役目も価値も半減してしまうのに，これがやめられない。プリントの山は通信のたびに積み重ねられ，しかたなく製本したら，2cmほどの厚さの本が60冊にもなった。

情報の圧縮と小型化のためのマシンが，半面，情報の拡大化をさせているこの矛盾を，どう解決するか，それが当面のわが家の課題になっている。

玉突きジャンプ

ライティングデスクがマシンルームの中央に置かれ，この中に98が収められて久しい。この98のキーボードの乗っているデスクの面が少し高すぎて，入力するときに腕の疲れが気になる。きょうは，これを改善するために，デスク面の下側に，スライド式のプレートを取り付けて，そこからキーボードを引き出して使えるようにしようという計画なのだ。

きのう，夫が東急ハンズから材料を仕入れてきた。左右一対のレール金具と，塗装した白い木製のプレート，引き出すための金属の取っ手。

デスク面そのものも，もとは開閉式の1枚のプレートだから，その下にもう1枚のプレートを取り付けるのは，構造的に不安もあった。でも，アイデア実現の夢は，とてもおさえられない。

朝食のあと，さっそくマシンルームに入る私たちは，ホンニャアもいっしょだということを，あまり気にとめなかった。

準備が整い作業が始まる。図画工作は私の専門だから，ノコギリ，ペンチ，ドライバーを見るとうれしくてたまらない。新しいしかけのコーナーをじぶんたちでこしらえる喜びで，ワクワクしてくる。

そして，順調に構想どおりのものができあがっていった。レールのすべりぐあいがとても良いので，水平に取り付けるために，とくに注意が必要だった。ボルトのしめつけにも，このごろは便利な工具があるのでありがたい。

ホンニャアの見学好きは子猫のころからだった。きょうは，X68000の収まったラックの頂上で，ウツラウツラとながめていたらしい。ドリルの音は昔から気に入らないのだが，電気ドリルではないのでガマンしていた。でも，なかなか終わらない。そろそろ別の部屋に移動しようかと立ち上がったとき，ちょうど中腰になった夫の背中があったのだ。

「文豪MINI7HR」は，単独でキャスター（小車）付きのラックに乗っている。これはキーボード用の台が折りたたみ式でついているが，だいたいはいつでも使えるように，台をひろげ，キーボードも乗せられている。

キャスターのおかげで，どこにでもころがしていける「文豪」が，今日は工作中の夫のすぐそばにきていた。

ホンニャアはいつもの気分で，夫の背中めがけてジャンプした。中腰の夫は，いきなりの重みに，バランスを失ってよろめき，「文豪」から突き出したキーボード用の台に接触した。

このときばかりは，キャスターがわざわいして，ラックは前倒れになり，「文豪」本体は，まずキーボードを直撃，そのあともちろん床まで落下した。ホンニャアの一撃が，玉突き状につぎつぎと事件をひきおこしたのだ。

1周年記念日

おそるおそる電源を入れた「文豪」のモニタは，「キーボードが正しく接続されていません」と表示されるばかりで，何も反応しなくなった。あれだけの墜落をしたのだから，モニタに文字が出るだけでもふしぎなくらいだった。

「文豪」に倒れられては一大事。こんなときは，「真光無線」のタカダ部長に頼るしかない。さっそくアキバのラジオ会館に本体とキーボードを運んだ。

さりげなく話を聞いてくださったタカダさん。「保証書が利くかもしれませんよ」。

まさか，「故障状況」の欄には「モニタ落下，キーボートの足破損，正常動作せず」と，堂々と書かれているのに。

数日後，タカダさんからのお電話。

「本体は異常なし。キーボードはユニット交換するそうです。それから，保証書が使えましたよ。1日違えばダメでした」

「文豪MINI7」から「MINI7HR」に買い換えた日から，丸1年目のできごとだったのだ。ひとつの創作から，ひとつの破壊をつくった日だったけれど，ラッキーな判定をいただいた日でもあった。

NEW PRODUCTS

高輝度液晶ビジョン
XV-H1/H1Z
シャープ

XV-H1

シャープは従来の液晶ビジョンより明るさを向上させた高輝度タイプの液晶ビジョン「XV-H1」,「XV-H1Z」の2機種を発売した。

「XV-H1」は液晶パネルとランプの改善により従来機「XV-100Z」に比べ2倍明るく，しかも固定短焦点レンズの採用により3mの投影距離で100型の映像を映し出すので（従来機では4.6m），6畳間でも大画面が楽しめる。

「XV-H1Z」は従来機比1.7倍の明るい映像を実現，また20型から100型までのサイズフリーで楽しめる1/2倍ズームレンズを搭載している。

どちらの機種も従来機比約80%の体積，重量というポータブルなものになっていて，入力はS映像端子×2，ビデオ×2が用意されている。また，設置時に面倒なコンバーゼンス（色合わせ）調整がいらないため，取り扱いがフォーカス合わせだけの簡単な操作になっている。さらに，映像左右反転機能が採用されていて，背面（透過）投影などにも対応している。

「大画面シアター」をトータル的に提案するクリアタイプの新スクリーン，AVサラウンドアンプ，スピーカーシステムなどのシステムアップ機器もあわせて発売。

価格は「XV-H1」が450,000円,「XV-H1Z」が500,000円（どちらも税別)。

<問い合わせ先>
シャープ㈱　☎03(260)1161, 06(621)1221

ビデオレセプター
XC-100P
シャープ

XC-100P

シャープは会議や講演会などで手元資料を拡大して映し出す，画像入力装置「ビデオレセプター」を発売した。

「ビデオレセプター」は平面原稿はもとより商品などの立体物もフルカラーで撮影し，液晶ビジョンやモニタテレビに映すことができる。画像は75万画素(25万画素×R,G,B)の高解像度を実現。1画面を記憶させ必要なときに引き出せる「画像コールバック機能」や，文字などを重ね合わせて映せる「スーパーインポーズ機能」なども備えている。

また，撮影画像にタイトルやグラフなどのパソコンで作成した画像をスーパーインポーズすることもできる。X68000シリーズのほか，水平同期周波数15kHz/15ピンアナログRGB出力のモードを持つパソコンが接続できる。パソコンとのシステム化で画像のファイリングなどが可能なようにパラレルインタフェイスが装備されていて，読み取り領域が容易に設定できるカラーイメージスキャナとしても使うことができる。映像出力はビデオ/S端子/RGBアナログ端子が用意されている。

価格は398,000円（税別)。

<問い合わせ先>
シャープ㈱　☎03(260)1161, 06(621)1221

電子手帳で測量
PTS-IIIシリーズ
旭精密

旭精密とシャープは電子測量機と電子システム手帳をシステム化し，旭精密ブランド（PENTAX）としてトータルステーション「PTS-IIIシリーズ」を発売した。

このシステムは従来，手入力あるいは高価な専用のデータコレクタを利用して行っていた測定データの入力を，自動的に電子手帳のカードに記憶させ，測量設計CADシステムに転送できるものである。

付加的な機能として，記録したデータを用いて座標計算を行い面積を求める計算機能もついている。

価格（いずれも税別）
トータルステーションPTS-IIIシリーズ
1,600,000/2,000,000円

ICカードと専用ケーブル

DT-21A/MB-65(66) 91,000(88,000)円

データコンバータ 105,000円

カードリーダライタ 40,000円

<問い合わせ先>

旭精密㈱ ☎03(593)0421

電子手帳で過去帳管理
お寺さんカード
シャープ

お寺さんカード

シャープと株式会社ぴっぱらは電子システム手帳で過去帳管理ができる「お寺さんカード」を開発，発売した。

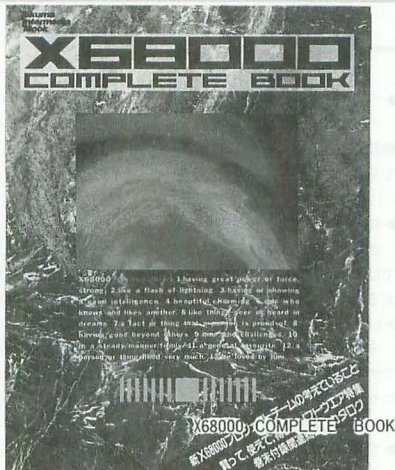
この「お寺さんカード」は壇信徒と正確な法事スケジュールを立てるために，膨大な過去帳を手帳方式にして，常に携帯でき簡単なキー操作で各壇信徒ごとのスケジュールが即座にわかる。

価格は30,000円(税別)。

<問い合わせ先>

㈱ぴっぱら ☎0742(27)1737

BOOK

トクマインターメディアムック
X68000 COMPLETE BOOK
徳間書店

徳間書店からX68000の本，「X68000 COMPLETE BOOK」が発売された。ツールやゲームなどさまざまなジャンルのソフト紹介，最新ハードウェア紹介，シャープ液晶映像システム事業部の岡本，金井両氏へのインタビュー，巻末付録関連商品大カタログなどから構成されている。

価格は1,200円(税込)。

INFORMATION

第7回 ホビーマイコンショウ

FORESIGHT10周年，きまぐれコンピュータクラブ10周年，FBI-NET5周年を記念してホビーマイコンショウが開催される。

FORESIGHT，きまぐれコンピュータクラブはTK-80やPC-8001ユーザーを母体とするパソコンクラブで，FBIは草の根BBSネット。

いずれもホビーとしてのマイコンを愛するクラブで会員の制作したハードおよびソフトをマイコンのメッカ，ラジカンで展示する。

おおいに，マニアの皆さんと交流したいとのこと。アキバショッピングのついでに立ち寄ってみては。

日時：10月14日(日) 11時～17時

会場：ラジオ会館8階大ホール

出品：X1,MSX2などのホビーマシンから最新のマシンまでを利用したソフト，ハードまで。また，パソコン通信の新システムなども。

〈問い合わせ先(事務局)〉

〒134 江戸川区西葛西5-7-8-503

峰岸 順二 ☎03(675)1964

第4回 全国(草の根)BBS大会

1.テーマ 趣味から実用へ向かうパソコン通信

2.開催日時 1990年10月21日 10時30分～16時30分

3.会場 東京：北の丸科学技術館2階 千代田区北の丸公園2-1 ☎03(212)8471

4.開催内容
○ISDNを使ったパソコン通信の講演と実演
○抽選会 豪華商品が当たるゲーム大会
○協賛企業，諸団体，関係官庁の展示・実演
○ＢＢＳへのフリーアクセスターミナル
○懇親会（ティーパーティ交歓会を会場で行います)
○オンライン3分間クッキング実演
○全国BBS連絡協議会総会

5.主催 全国BBS連絡協議会

6.協賛 パソコン通信関連ソフトハウス・メーカー・諸団体，出版社，商用ネットワーク

7.事務局 ㈱新企画社 ポプコム編集部 大藤 謙二 ☎03(263)6940

Oh! X & Oh! FM
バックナンバーフェア

Oh!X，Oh!FMの創刊号から最新号までのバックナンバーが特設コーナーで展示販売される。少数ずつながら，いまや入手しにくいものまで一堂に展示される予定。。

場所：書泉グランデ 5F
東京都千代田区神田神保町1-3-2
☎03(295)0011

期間：9月20日(木)～10月20日(土)

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。暑かった夏ももう終わり，秋の行楽シーズンの到来です。さあ，何して遊ぼうかな（実は年中遊んでいたりして……）。

一般

▶今こそパソコン通信を始めよう
パソコン通信をめぐる現状の紹介に始まって，必要な機器，主なネットやパケット通信サービスなどの体系まで掲載している。——山田憲一，マイコン，9月号，114-133pp.

▶MUNEPI♪のゲーム探検隊
IBM用の「STARWARS」を紹介。ベクタースキャン方式ではないけれど，ゲーセンの興奮がAX286などでも味わえるぞ。——MUNEPI♪，マイコン，9月号，204-205pp.

▶LET'S PROGRAM
今月の宿題発表は「3×3の魔方陣の解法」。X-BASICでの解答のほか，X1マシン語で4×4の魔方陣の組み合わせを計算するプログラムも寄せられている。——藤本健，マイコン，9月号，224-232pp.

▶やまさんのアルゴリズムブック
データ処理の一環として，リスト処理について考える。リストの基本的な構造，データの追加処理について。——やまさん，マイコン，9月号，304-305pp.

▶実践ハード入門
今月は液晶ディスプレイを使ってのメッセージボードの製作にチャレンジ。液晶の仕組み，液晶ユニットの使い方についても解説する。——石川至知，マイコン，9月号，309-313pp.

▶マイコン認定試験受験講座
7月に行われたマイコン認定試験の4級・3級の全問題と解答例を掲載。——マイコン認定試験受験研究会，マイコン，9月号，382-399pp.

▶失敗しないプリンタ選び'90
プリンタに関する基礎知識，カタログ用語の解説に関する文章と，熱転写，インパクト，インクジェット，ページの各方式の代表機種の紹介・比較記事が掲載されている。——編集部，ASCII，9月号，226-248pp.

▶MEDIA BREAK
CESにおいて発表されたCommodore社のCDTVは，Amiga 500にCD-ROMドライブを組み合わせてシェイプアップしたものである。その狙いと今後の計画について開発責任者にインタビューする。——編集部，ASCII，9月号，362-363pp.

▶ワン・ステップ通信
ゲームメーカーとして知られる株式会社ナムコの福祉機器分野への取り組みについてレポートする。——編集部，ASCII，9月号，380-381pp.

▶なんでもQ&A
All in Noteのソフト，ハードに関する4つの質問に答える。同梱ソフト以外にどんなソフトが使えるか，辞書ROMの用途は，などなど。——編集部，マイコン，9月号，412-413pp.

▶田原総一朗のコンピュータ・ルポ
ノートパソコン"All in Note"を発売し，AX互換路線における本格的な展開を開始したシャープ。このAX路線に賭けるシャープの意気込みとパソコン再生構想を，コンピュータ事業部長の西岡郁夫氏に伺う。——編集部，THE COMPUTER，9月号，121-131pp.

▶ネットワーカー・ホリック第27回
PDSが，ウイルスの標的となっていることを述べ，ダウンロードする際の注意点，主なワクチンの入手法を紹介。——編集部，LOGIN，16・17号，254-255pp.

MZシリーズ

MZ-700/1500(Hu-BASIC)

▶KABE '89
迫りくる壁の間を逃げ回るゲーム。——江崎実，マイコンBASIC Magazine，9月号，170p.

MZ-1500

▶都市計画
シムシティーを意識した都市シミュレーションゲーム。——山野辺太郎，マイコンBASIC Magazine，9月号，128-130pp.

MZ-2500(BASIC-M25)

▶ニトラス
じっくり考えるブロックパズルゲーム。——アラケン&ken/ichiro，マイコンBASIC Magazine，9月号，131-133pp.

X1/turbo/Z

X1シリーズ

▶WING X
ゼロ戦で敵戦艦に特攻する。——菊池光吉，マイコンBASIC Magazine，9月号，158-159pp.

▶MICRO WARS
細胞内が舞台のシューティングゲーム。——栗林良樹，マイコンBASIC Magazine，9月号，160-161pp.

▶誌上公開質問状
X1Gモデル10に接続できるフロッピーディスクの種類に答えている。——多田太郎，マイコンBASIC Magazine，9月号，92p.

X1+FM音源ボード（要NEW FM音源ドライバ）

▶ナイトストライカー
タイトーのナイトストライカーより「アーバン・トレイル」のミュージックプログラム。——Com，マイコンBASIC Magazine，9月号，185-187pp.

X1turboシリーズ

▶How To Win

参考文献
I/O　工学社
ASCII　アスキー
コンプティーク　角川書店
THE COMPUTER　ソフトバンク
テクノポリス　徳間書店
POPCOM　小学館
マイコン　電波新聞社
マイコンBASIC Magazine　電波新聞社
LOGIN　アスキー

新刊書案内

本書のタイトルがライフゲームではなく「ライフゲイム」というのは，訳者のこだわり。なかなかこだわる訳者で，「マックスウェルの悪魔」といわずに，「マックスウェルのデモン」という。

原題を訳すと「再帰的な宇宙」。これはライフゲームの織り成すさまざまなパターンの考案から，フォン・ノイマンが研究した自己再生オートマトンなどを経て「ちょうど自己再生する動物のようなふるまいをするライフゲームのパターン」へと至るライフゲームの奥の深さを楽しむ本なのだ。なかにはビッグバンやマックスウェルの悪魔などなどさまざまな話がからんでくる。

ライフゲームでは驚くほどのパターンが現れ，そのほとんどは研究されて名づけられている。形状から名づけられた船やカヌー。一定のパターンで形を変えながら移動していくグライダー。グライダーに出会うとそれを破壊しながら自分は無傷なイーター。ライフゲームのアルゴリズムは極めて簡単であり，BASICですぐ組める程度のものだ。本書には試してみたい人のためのBASICとアセンブラ（共にIBM-PC用）も掲載されている。（K）

ライフゲイムの宇宙　ウィリアム・パウンドストーン著　有澤誠訳　日本評論社　☎03(987)8611　A5判　275ページ　3,800円

三國志II，セレクテッドソーサリアン5の紹介。——編集部，コンプティーク，9月号，132-135，148-149pp.

▶ソフトレビュー

セレクテッドソーサリアン5の紹介。——編集部，コンプティーク，9月号，33p.

▶NEW SOFT

セレクテッドソーサリアン5，ミスティVol.4の紹介。——編集部，LOGIN，16・17号，14-15pp.

X68000

▶X68000マシン語入門

今月まで3回にわたって掲載してきたプログラムの詳しい解説を行う。グラフィック関係のシリーズの総まとめ。——高橋雄一，マイコン，9月号，356-364pp.

▶なんでもQ&A

Hyperwordで外字を使うには，Cコンパイラのエラーファイルを出力できるか，Communication PRO-68Kの概要は，などの質問に答える。——編集部，マイコン，9月号，410-411pp.

▶AVプログラミング講座

アナログジョイスティックのハード面，ソフト面からの解説と，IOCSを使ったサンプルゲーム。スプライトプログラミングの項ではスプライトの基礎的な知識について解説する。——中山進・仲田津宏，ASCII，9月号，289-296pp.

▶zboot.sys

X68000のSRAMを利用したブートデバイスセレクタ。複数のOSで複数のパーティションを設定したハードディスク，SRAM上のプログラム，ROMなどからブートすることが可能。——中山進，ASCII，9月号，317-318pp.

▶ps.r

現在メモリ上にロードされているプログラムのリストをまとめて報告するコマンド。PROCESSとは違ってフックされているベクタの表示も行える。——中山進，ASCII，9月号，319p.

▶AV STRASSE

S映像出力対応の低価格ビデオボードCZ-6BVI，バージョンアップされたCommunication PRO-68K，ポリゴン処理のフリーウェア「CAPTAIN」などが紹介されている。——編集部，ASCII，9月号，337-340pp.

▶長期ロードテスト

Human68k ver2.0とASK68K，日本語ワープロを試用する。ASK68Kに関しての評価は高いが，カスタマイズを無視するワープロ，「かな」「ローマ字」などのキーが遠い点に関しては厳しい見方をしている。今月は17時間使用。——編集部，ASCII，9月号，384-392pp.

▶GAME BOX

「クォース」「ダウンタウン熱血物語」の紹介。——編集部，I/O，9月号，121-127pp.

▶Small-Tool

Human68k用の便利な命令集。属性変更・ワイルドカード対応DUMP・OPMDRV解除・コントラスト調整・ウエイト・ドライブ制御の6つのコマンドが掲載されている。——曇りときどき晴れ，I/O，9月号，155-163pp.

▶CRTDRV.X

X68000の画面の色を手軽に変更するためのプログラム。定義ファイルでカスタマイズも可能。——高橋美幸，I/O，9月号，168-173pp.

▶プリンタデータ・キャッチャ

プリンタに対して送られるデータをファイルに落とすプログラム。グラフィックを加工して何度もプリントアウトを行う場合などに便利。——市原昌文，I/O，9月号，174-176pp.

▶PCG SAVER

PCGの内容をセーブする。市販のゲームのキャラクタなどを使用したいときに使おう。——Soft工房の職人Sugar，I/O，9月号，177-179pp.

▶SOFT BOX

シャープからリリースされた「Communication PRO-68K」の新バージョン"2.0"を紹介。——桜井智史，I/O，9月号，224-225pp.

▶SOFT BOX

単体で発売になった「SX-WINDOW」の概要を紹介。——川田義治，I/O，9月号，229p.

▶E_PRINT.X

エラー出力をリダイレクト機能を使ってファイルに結果を残すために，エラー出力を標準出力にしてしまうユーティリテイ。——L&M，I/O，9月号，244-245pp.

▶TRON

2人で遊べるゲーム。1ドットを見極められるか。——高橋潤，マイコンBASIC Magazine，9月号，162p.

▶ボンバータコくん

時限爆弾を仕掛けて，その爆風でモンスターを倒すゲーム。——高橋秀之，マイコンBASIC Magazine，9月号，163-165pp.

▶先取りおすすめゲーム

クォータースタッフ，銀河英雄伝説IIを紹介。——編集部，テクノポリス，9月号，6-11pp.

▶GAMING WORLD

発売予定の麻雀ゲーム「RYU～哭きの竜」を紹介——編集部，テクノポリス，9月号，22p.

▶新作ゲーム先取り！ Soft Flash

新作ゲーム「ペルセウスの冒険」を紹介。——編集部，テクノポリス，9月号，28p.

▶How To Win

ウルティマVの紹介。——編集部，コンプティーク，9月号，140-143pp.

▶X68000 SPIRITS

トンネルズ&トロールズ，幻獣鬼，プロテニスワールドコート，バルーサの復讐，ジェミニウイングなどの新作ソフトを紹介。——編集部，コンプティーク，9月号，240-243pp.

▶最新ソフト徹底紹介

ウルティマVの攻略法を紹介。——編集部，コンプティーク，9月号，86-87pp.

▶WE ARE THE X68000 WORLD

新作，制作中のソフトを紹介。幻獣鬼，ザ・マジカルショット，プロテニスワールドコート，F.S.S.ペルセウスの冒険。——編集部，コンプティーク，9月号，90-92pp.

▶新作HOT情報

シムシティーを紹介。——編集部，POPCOM，9月号，20p.

▶ミュージックパビリオン大全集

おどるポンポコリン，ゆめいっぱいのミュージックプログラム。——PONPOKORIN GOTOH，POPCOM，9月号付録2，2-8pp.

▶NEW SOFT

第4のユニット第5弾「D-Again」を紹介。——編集部，LOGIN，16・17号，27p.

▶X68000新聞

イメージファイト，AXIS～FZ戦記～，幻獣鬼，三國志II，提督の決断，第4のユニット5・D-Againを紹介。——編集部，LOGIN，16・17号，180-183pp.

▶最新ゲーム徹底解剖

シムシティー，ウルティマV，ギャラガ'88を徹底解剖する。——編集部，LOGIN，16・17号，212-215，228-231，240-241pp.

ポケコン

PC-E500

▶縞・縞

自分と敵が表示されないゲーム。敵は自分の足跡を消していくから，そこから敵の動きを推理して，捕まらないように画面上のドットを拾う。——Calling，マイコンBASIC Magazine，9月号，167p.

▶AGOUKの野望2

死んだはずのAGOUKが再び攻撃してきた。——森高周作，マイコンBASIC Magazine，9月号，168-169pp.

PC-1500

▼誌上公開質問状

PC-1500でPC-1245用やPC-1251用のプログラムが使えるか。また，PC-1245用のプログラムをPC-1246DBやPC-1248DBで使えるか，などに答えている。——Akiko，マイコンBASIC Magazine，9月号，91-92pp.

コンピュータソフトマニュアルの書き方

最近，ドキュメントの書き方に関する本が増えてきた。これもそのひとつ。原本はAPPLE IIのユーザー教育グループのために書かれたもの。APPLE IIのマニュアルといえば当時，手本とされるほどできがよかった。本書も読みやすく，具体的によいマニュアルを書く方法について述べている。訳はひどいが本書に従って書くことができればもっとましなマニュアルが増えるだろう。　（K）

ジョナサン・プライス著　辻新六監訳　ユーザフレンドリー研究会訳　日刊工業新聞社　☎03(222)7111　A5判　340ページ　3,000円

ザ・ゲームカタログ'90

光栄の書籍だからといってパソコンゲームとは限らない。本書は大人でも遊べるゲームを集めたカタログである。一度に遊べる人数で区分けされており，カタログとしては非常にみやすい。パソコンゲームやエッセイ，モノポリー名人戦誌上再録ものなどなどバラエティは豊富だが，一番楽しいのはカラーで輸入もののボードゲームを紹介したカタログページだ。特にボードゲームのセンスあるゲーム盤は見ていて飽きない。　（K）

光栄編集部編集　JAGA監修　光栄　☎045(561)6861　A4判変形　157ページ　2,800円

QUESTION and ANSWER

X1turboZとX68000のユーザーです。CZ-8PC4を使って1/4の大きさの48ドットカラーハードコピーを取りたいのですが，残念ながらマニュアルには，1/4の大きさのハードコピープログラムが用意されていません。そこで，Oh!MZ1987年9月号掲載のCZ-8PC1用のダブルサイズのハードコピープログラムをうまく改造すれば，48ドットプリンタでは1/4の大きさになるのではないのかと考えたのですが，素人の浅知恵でしょうか？　京都府　福知　健

モノクロプリンタでカラー画面のハードコピーを取りたいのですが，カラー画像をモノクロ画像に変換するにはどんなやり方があるのですか。　埼玉県　菅野　公紀

質問の意図は1/4の大きさのプリントを取るというよりも，4倍密度のプリントアウトを行いたいということだと思います。

プリンタの制御コードだけ見れば24ピンビットイメージ印字を行うか48ピンビットイメージ印字を行うかの指定は簡単に変更できますが，実際に印字を行うには一度に48ドット×横幅分のデータをセットしておかなければなりませんので，単純な変更ではだめでしょう。

48ピンビットイメージプリントを行うために必要なことは48ドット幅のパターンデータを用意して，48ドットイメージ印字命令とともにプリンタに送るだけです。印字命令についてはプリンタのマニュアルにあり，速度さえ無視すれば操作はBASIC上でも可能です。まず，プリンタマニュアルのサンプルなどを見て用意したデータがどのように印字されるかを実際に確かめてみてください。

次に菅野さんの質問に答えましょう。モノクロでの表現力は白か黒かに限られています。しかし，それだからといって馬鹿にしてはいけません。身近なところには新聞に印刷されている写真などがあります。これらは色の濃淡で像を表現していますが，これでも十分に綺麗な像が得られることがわかると思います。

昔からカラー画像をモノクロプリンタに打ち出すときによく使われる方法に，ピクセルのカラーコードを複数の点で表すものがあります。この複数の点の集まりをマトリクスと呼びます。3×3のマトリクスならその中に点が何個入っているかによって10個のマトリクスパターンを作れることになります。つまり10段階の濃淡表現ができるわけです。

あらかじめ画面の表示色に対応したマトリクスパターンを用意しておき，カラーコードを対応するマトリクスに置き換えることでモノクロ変換するのです。これは簡単でわかりやすいのですが，変換後の画像が元絵より大きくなってしまうことが欠点といえば欠点です。また，自然な感じのマトリクスパターンを作ることが大変だし，元絵の色数が多い場合はお手上げの状態になります。

元絵の大きさが変わらず，しかも，かなり良質のモノクロ画像が得られるアルゴリズムが本誌1988年11月号に桒野雅彦氏によって発表されています（最近，誌上で引用されることが多いですね）。このアルゴリズムの基本的な考え方は，色を明るさで考え，その明るさの総和を等しくしようというものです。いま黒の明るさを0，白の明るさを1として，明るさ0.7の点と0.2の点が横方向に交互に8個並んでいるデータをモノクロに変換することを考えてみましょう。

まず最初の点を見ると明るさは0.7です。これは白の明るさである1を満たしていません。ですからこのドットは黒にして，ひとつ右にデータとして0.7を送ります。2つめは0.2です。これに送られてきたデータ0.7を足すと0.9になります。ここも黒です。データとして0.9を送ります。

しかし，3つめのデータは0.7に送られてきた0.9を足して1.6，ですから白です。次に送るデータは白（1.0）を引いた0.6になります。4つめは0.2+0.6=0.8。だから黒。0.8をデータとして次に送ります。5つめは0.7+0.8=1.5。白。次に送るデータは小数点以下の0.5です。6つめは0.2+0.5=0.7。だから黒。送るデータは0.7。7つめは0.7+0.7=1.4。つまり白。送るデータは0.4。8つめは0.2+0.4=0.6。すなわち黒。変換後は白が4つ，黒が4つとなります。

明るさの総和は0.7×4+0.2×4=3.6ですから，白が3つ黒が5つでだいたいよいことになります。ここでは右にしかデータを送っていませんが，実際には輝度データを4分割して左下，真下，右下にもばらまいています。

カラーコードを明るさに変換するには，

0.30R+0.59G+0.11B

を計算します。8色ならRGBは0か1です。黄色ならR=1，G=1，B=0ですから明るさは，0.30+0.59+0=0.89となります。RGBをすべて1にして計算すると1になります。これが白の明るさです。

これでカラーグラフィックをモノクロデータに変換できます。マトリクス式の変換では，どうしてもマトリクスの大きさで変換像の大きさが制限されることが多いのですが，桒野式ではちょっとした処理を加えることでドット単位の大きさ変更も可能です。プリンタへの出力にはもってこいのアルゴリズムといえます。　（影山　裕昭）

X68000EXPERTIIを購入して4カ月になりますが，N88-BASICのWINDOW文，VIEW文にあたる命令はX-BASIC(Version 2.0)にはないのでしょうか？　もしなければ，WINDOW文，VIEW文と同様の機能を実現する方法を教えてください。　神奈川県　小川　敏昭

Oh!Xの読者の方はN88-BASICといわれてもピンとこないと思いますので簡単に説明しておきます。PC-9801はグラフィック画面に対して3つの座標系を持っています。以下，それぞれの座標系について説明していきましょう（なお，以下の説明はPC-9801RXのマニュアルを適宜引用させていただいてます）。

○オリジナル座標系

この座標系は，常にディスプレイ装置と結びついた座標系です。すなわち，この座標系の大きさは，ディスプレイ装置と対応しており，座標上の各点が画面を分解するドット数と一致しています。X68000のスクリーンモード0では（0，0）-（1023，1023）を四隅とする座標系であり，スクリーンモード1および2では，（0，0）-（511，511）を四隅とする座標系です。

○スクリーン座標系

BASICの命令を使って描いた図形を表示する際には，ディスプレイ画面の中で表示領域を限定することができます。その限定した領域をスクリーン画面とか，ビューポートとかいい，そこでの座標系をスクリーン座標系といいます。このスクリーン座標系を設定する命令文，VIEW文を使用

して，

VIEW (100,100,200,200)

とすれば，オリジナル座標系にして (100,100)－(200,200) の範囲に，(0,0)－(100,100) の大きさのスクリーン座標系をとることができます。

○ワールド座標系

この座標系はディスプレイ画面に対応したものではなく，いわば論理的な画面です。PC-9801の場合，BASICでの描画命令PSET文やCIRCLE文は，この画面に対して描画を行います。ここに描かれたものが，スクリーン画面に投影されると考えます。

普通，ワールド画面は，途方もなく広いので，(今回の場合，縦横ともに，－3.5953862697246E＋308から3.5953862697246E＋308までとれる)，画面のすべてをスクリーン画面に投影していたのでは，見にくくてしょうがありません。B5判の東京都全道路地図でも考えてください。こんなもの，使い物にならないでしょう。こんなわけで，もの凄い縮小率になるのを避ける方法があります。

スクリーン画面に投影してやる範囲を，限ってやるのです。(東京ロードマップだったら，池袋駅周辺だとか，高田馬場駅周辺だとか) その限定された領域をウィンドウ領域といいます。前述の東京ロードマップだったら，ウィンドウ領域に，高田馬場駅周辺を指定してやると，手帳サイズの高田馬場駅周辺の地図になるわけです。

そのウィンドウ領域というものが，WINDOW文で指定できます。その領域内に対して行う操作(線を引くだとか，円を描くだとか)は，すべて，VIEW文で指定されたスクリーン画面に，移されてしまいます。

さて，これらの概念がX-BASICでもサポートされているかどうか，ということですが，残念ながらX-BASICでは，完全にはサポートされていません。一応，論理画面，実画面，表示画面といいう分け方はあるのですが，表示画面は論理画面の中央に固定されているので，座標系の変更などはできません。

具体的にプログラムを組んだのが，リスト1です。方針としては，線や円を描く前にそのパラメータを横取りして画面操作の前に簡単な1次変換してやります。そしてその変換をしたあとに，描画を行います。行番号10～60，1000，1490が私の作った関数で，行番号70からが，サンプルプログラムです。関数「__line」は，プログラム中で私が定義したVIEW文に対応している関数です。X-BASICに備わっている関数「line」と取り替えてみると，その動作の違いがよくわかると思います。

行番号10，20，30で宣言されている変数は，それぞれ，あとの処理で使うグローバル変数ですので，貴方の作ったプログラム内で同じ変数を使用しているようならば，変数名を変更してください。

使い方ですが，まず，プログラムの先頭に，リスト1の10行から60行までをコピーします。そして，どこでもよいですから，1000行から，1250行までをコピーします。そうして，グラフィック描画関数の前にアンダーバー‘__’をつけます。今回サポートした関数は以下のとおりです。

__view，window，line

__circle (ただしパラメータはx,y,pの3つだけ)，__box

ほかのものはこれを参考にすれば簡単に作成できるでしょう。あ，そうそう1060行，1070行の511という数字は，このサンプルプログラムでのスクリーンモードの表示画面サイズ (512×512) からきているので，ほかの画面モードを使うときはここの値も適当に変えてください。　(石上　達也)

リスト1

```
  10 float __mx,__my                            /*縦横の倍率
  20 float __wx1,__wx2,__wy1,__wy2
  30 int   __sx1,__sx2,__sy1,__sy2
  40 __wx1=0:__wy1=0:__wx2=511:__wy2=511 /*初期値設定
  50 __sx1=0:__sy1=0:__sx2=511:__sy2=511 /*スクリーンモードで変えて下さい
  60 /*
  70 screen 1,1,1,1
  80 _window(0,0,639,399)
  90 _view(0,0,511,511)
 100 _box(0,0,639,399,7)
 110 for x=0 to 128
 120   _line(x*5,0,639-x*5,399,3)
 130 next
 140 end
 150 /*
1000 func _window(x1;float,y1;float,x2;float,y2;float)
1010   __wx1=x1:__wy1=y1
1020   __wx2=x2:__wy2=y2
1030 endfunc
1040 /*
1050 func _view(x1;int,y1;int,x2;int,y2;int)
1060   if x2>511 then x2=511             /*スクリーンモードによって変えて下さい
1070   if y2>511 then y2=511
1080   window(x1,y1,x2,y2)
1090   __sx1=x1:__sy1=y1
1100   __sx2=x2:__sy2=y2
1110   __mx=(__sx2-__sx1)/(__wx2-__wx1)
1120   __my=(__sy2-__sy1)/(__wy2-__wy1)
1130 endfunc
1140 /*
1150 func _line(x1;int,y1;int,x2;int,y2;int,p;int)
1160   line(__sx1+(x1-__wx1)*__mx,__sy1+(y1-__wy1)*__my,__sx1+(x2-__wx1)*__mx,_
_sy1+(y2-__wy1)*__my,p)
1170 endfunc
1180 /*
1190 func _circle(x;int,y;int,r;int,p;int)
1200   circle(__sx1+(x-__wx1)*__mx,__sy1+(y-__wy1)*__my,r*__mx,p,0,359,256*__my
/__mx)
1210 endfunc
1220 /*
1230 func _box(x1;int,y1;int,x2;int,y2;int,p;int)
1240   box(__sx1+(x1-__wx1)*__mx,__sy1+(y1-__wy1)*__my,__sx1+(x2-__wx1)*__mx,__
sy1+(y2-__wy1)*__my,p)
1250 endfunc
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒108 東京都港区高輪2-19-13
NS高輪ビル
ソフトバンク株式会社出版部
「Oh! X質問箱」係

FROM READERS TO THE EDITOR

暑い，暑いといっているうちに，すっかり，いまはもう秋になってしまった。ああ，涼しいなあ。本当に本当に涼しいなあ。うそじゃないよ。こんなに涼しいと思わず月見にでも行ってしまいそうになるなあ。ああ，涼しい。

◆先日，下のチビ（我が子のことです）があんまり泣いて駄々をこねるので，「怪獣を見せてやる！」といってドラゴンスピリットをやらせたら，「ボボーッと火を吹くよお！」と夢中になってしまいました。それから，毎夜「怪獣を見にいこう！」と親の私の手を引くのでした。ちなみに，もうじき3歳になりますが先が思いやられるのでした。くわばら，くわばら……。

伊沢　範庸(30)東京都

そういう場合は見るも耐えないようなつまらないゲームを見せるとか，聞くに耐えないようなものすごい音楽を鳴らすようにすると，二度とコンピュータに近寄ろうとしないでしょう。

◆コンピュータが登場する映画を古いものから順に紹介する，という企画はいかがでしょうか？　コンピュータ≒ロボットというところでしょうか？　「禁断の惑星」のロビー，「2001年宇宙の旅」のHAL etc.。

佐藤　信一郎(32)東京都

そういや結構いっぱいありますよね。「メトロポリス」のロボットのマリアとか，「審判」に出てきた古めかしくてやたらでっかいコンピュータとか。でも，いちばん古いのはどの映画だろう。

◆ポケコンの連載が開始されることなど久しくありませんでしたね。これからも定期的な掲載を希望しております。それから，Oh!X LIVE in '90には硬派なR&Rを求めています。ブリティッシュビートかグラムロックを載せていただきたい！

野口　穂積(22)大阪府

載せていただきたいといわれても投稿が来ないとね。僕としてはアル・ディ・メオラの「スペイン高速悪魔との死闘」とか，イングヴェイ・マルムスティーンの「トリロジー・スーツ OP:5」とかを送ってきてはしいな。

◆編集部ではいつも対戦ポピュラスをしたり，YETのスコアを競ったりしているのですか？

山中　政宣(16)三重県

いや，そんなことばかりしていると疲れちゃうので，たまには息抜きに仕事もしていますよ。

◆2カ月前のディスクをやっと解凍しています。最初はわからず子供に話していました。電子レンジでチーンしようか，水に浸けて解凍しようか。子供に笑われました。5年たっても初心者です。

矢野　博志(46)福岡県

いやあ，2カ月もたったから自然解凍してしまったんですね。ああ，よかった，よかった。

◆浪人してとうとう前期が終わってしまった。あっという間に過ぎていってしまった感じだった。3年の頃はあんなにのんびりしてたのに……。大学に行った高校の友人に会うとなんか劣等感を感じてしまう自分が情けないです。編集部の人やライターの人はやっぱり浪人した人はいないのでしょうか。

木下　卓也(18)埼玉県

いや，結構いますよ。うーん，それにしても浪人というのはやっぱり変なプライドを持っているものだと思うんですけど。そのへんの大学生よりも賢いんだと言い張るとか。そうでもしないと浪人なんかやってられませんよ。

◆ハードディスクつきのX68000を買ったのはよかったが，3DKの男3人住まいではブレーカーが飛ぶ飛ぶ（笑）。おかげで怖くて80Mバイトが使えないのであった！　悲劇！

鳥居　伸英(20)愛知県

ブレーカーって電子レンジが「チーン」といった瞬間とかに，よく落ちますよね。昔，うちの家も「チーン」といった途端に部屋が真っ暗闇になったりしましたものです。しみじみ。

◆わっ，私はとうとう愛機X1turbo model10を裏切ってしまった。そう私はついにあのX68000 EXPERTIIを買ってしまったのだだだっっっ。しかし，その代償はあまりに大きい。まず何よりも車の保険と合わせて月3万円×男の30回払い！　そして，いままで使用していたCZ-8PK2というプリンタがX68000では使えないのであった。X1turboの怨念か？

野村　耕嗣(19)千葉県

かもしれない。ところで，男の30回というのはよく音楽雑誌に載っているどっかの楽器屋の広告の「男の120回払い」の真似なんでしょうね。あの広告は最初見たとき，結構笑えました。

◆ポピュラスの世界には女性がいないのに，どうして人口が増えるのだろう。

影山　秀和(18)広島県

そういや，そうですね。やっぱり，細胞分裂かな。

◆私のX68000が死んだ。IPLのROMを取り替えてもらったけれど，直らず，再修理。今度は行ったきり帰ってこない（1カ月経過）。ディスプレイとキーボードが残った机を見た友人は「おっ！　ワークステーションみたいじゃん！　かっこいい！」とバカにする……ううっ……。ちゃんと直って早く帰ってきておくれ。うるうる……。

松尾　和浩(29)新潟県

そういう場合は，「いいだろー，カチカチカチ」とか口で言いながらキーボードを叩いてごまかすんですよ。で，相手がしらけちゃったらこっちのものです。ん？

◆はっきりいって，X68000はマウスがないとただの箱になってしまう。だから，マウスがなくてもカーソルを動かせるような機能がほしい

◀清水　健太郎　静岡県
不敵な笑みを浮かべながら，剣を振る。ちょっと，ヒーロー役というよりもかっこいい敵役という感じですが，いいですね。

◀白井　崇夫　神奈川県
自分でやらし〜絵だとか書いてますが，ぱっと見ると別にやらしくはないと思うんですけど。じっと見てると……。

(Amigaにはあったような気がしたけど)。つまり，OPT.1を押しながらカーソルを押すとマウスカーソルが動いてXF1，XF2キーがマウスのボタンに対応するとかしてほしい。これはただ単にアフターバーナー，ダンジョンマスターでマウスが危なく，それでいて金がないからだったりする。　浅沼　博明(20)北海道

うーん，あったらいいかもしれない。でも，その場合，つぎはキーボードがこわれてしまってもっとお金がかかるかも。

◆メガドライブの大戦略を買った。X68000版よりも面白い。船はあるし，MUSICもあるし，戦いもアニメしている。これでマウスが使えたら……。もし，X68000に大戦略IIを出すのなら，メガドライブ版を超えたものがほしいと思いませんか？　大戦略IIIもほしいけど。

中村　岳夫(17)東京都

そうか，やっぱりマウスが使えないというのが……。ところで，スーパーファミコンにもポピュラスやシムシティーとかダンジョンマスターが出るみたいだけれども，全部ジョイパッドでやるんでしょうか。指がつりそう。

◆今度のディスクの付録には，皆さんの声をサンプリングして入れてみてはどうでしょうか。もちろん，容量が余ったときにです。でも，ディスクのほとんどが編集部の会話の声だったら面白いかもしれませんね。宮浦　慎司(17)香川県

編集部の会話を入れたりするとほとんど僕の声しか聞こえないかもしれない。

◆部屋の壁いっぱいに国土地理院の5万分の1地形図を張り付ける。布団を敷いて電気を消す。暗闇に目が慣れた頃に壁を見る。うーん，ランドサットの気分（いま夜だったりする)。

松浦　範明(16)広島県

寝つきのいい人だと目が慣れる前にすっかり寝てしまったりして。

◆いやあ，とうとう100号ですか。早いものですね。あっ，はじめまして。私，小島というものです。これからももっと面白い記事を書き続けてくださいね。P.S.私，聞きましたよ。Oh!MZが出始めた頃，表紙のねーちゃんがいやらしいので，なんと“Oh!マゾ”と間違えられてエッチな本といっしょに並んでいたということを。

小島　勇人(18)青森県

そうか，そんな恥ずかしい過去がこの雑誌にはあったのか。

◆私の職場はF社のパソコン一色である。しかも，ワープロとして使用するだけである。なぜなら，MS-DOSもあまり知らない職員ばかりだからだ。その中で私は念願のX68000を買った。そこで職場の連中にX68000を買ったぞ！　というと，それなに？　PC-9801なら知ってるときた。頭に来た私は買ったばかりのX68000とシンセサイザで買ったばかりのソフトを使ってゲーム，グラフィック，MIDI演奏をしてみせてやった。それ以来，X68000の知名度はもとよりパソコン買うならX68000！　と株が上がったよ。パソコン＝ワープロというわが職場の考えも多少崩れたようだ。もちろん，本は薄くても中身は濃いと定評のあるOh!Xも宣伝しましたよ。　内匠　勇二(31)和歌山県

やっぱり，メジャーだという点ではPC-9801には大幅に負けていますからね。こういう草の根運動で輪を広げていきましょう？

◆暑いですね！　ところで，パソコンの冷却ファンから冷風が出るようになりませんか？　あれば，ぜひハードウェア工作入門で取り上げてください。　水谷　潔(43)三重県

簡単じゃないですか。まず，氷を手に入れます。そして，それを冷却ファンの前あたりに置きます。ほら，涼しい風が。ちょっと，湿気が多くなって体がべたつくのと，機械に水がかかるのが怖いけど……。

◆いったいどうしたら，ハミダシに載せてくれるんでしょうか。　松尾　保孝(18)奈良県

そうですね。しいていえば，まず短くまとまっていること。そして，なによりも面白くなければね。あっ，それともうひとつハガキを早めに出すことも必要です。

◆駅前の本屋でOh!Xがあったので（1冊きり！）ジュースを我慢して買った。しかし，あとで後悔した。家に着くまでノドはカラカラ。きわめつけは汗。さあ読もうというときになって表紙のきれいなCGの上に4本の指の跡がくっきり。読む気がまったくなくなった。

渡辺　安唯(16)千葉県

さあ，塗り絵の時間です。絵の具を買ってきて，自分で表紙に色を塗って直しましょう。

◆7月19日，AM10：20。頼んでおいたT&Tが届いた。Oh!Xの8月号を見たときは，やっぱり遅れているんだなあと思っていたが今日届いた。これって本当に12枚組だったんですね〜え。数年前（X1turboが出た頃）は，たしかフロッピーディスク10枚で7,000円ぐらいしていたと思うんですが……。すごいなあ。

川口　哲也(23)岐阜県

やっぱり，12枚組っていうのは多すぎますよね。手が12本ぐらいあれば全然問題ないかもしれないんですけどね。でも，出ないよりはまし？

▲大村　直人　北海道
似ているかどうかは……ですけど，「想像だけで書いたわりには」ってやつですね。本人はこれを見てどう思うんだろう。今度聞いてみよう。

▲姉帯　寛　宮城県
もう終わってしまったけれども，夏の名残に夏らしいイラストを。しかし，今年は結局，海には行かなかったなあ。

◆Kamikazeを使って1学期の成績をつけたら，「先生は冷たい。俺たちを機械で分別するのか」と詰め寄られた。でも，人間的に成績をつけるってよくないよ。ついつい好き嫌い（ひいき）が出ちゃうからね。だから，これからもKamikazeでクールに成績をつけるつもりだけど，「先生が使っているX68000って，でっかいファミコンみたいなもんでしょ」はないと思う。おこるよ，ホントに。　村井　裕弥(32)東京都

データを入力するときにひいきが出ちゃったりして。そんなことはないか。

◆“SWORD”を超えたS-OSといえば，その名前は“PEN”しかないだろう。

根内　賢一(19)千葉県

うまい！　ざぶとん1枚やっとくれ。

◆何年振りかで海に行った。そして，足が届かないところまで泳いでいったら溺れかけた。自然に対して人間の力というものが，いかに小さなものなのかということを思い知らされた1日でした。　千葉　浩貴(17)宮城県

うーん，ただ単に泳ぎが下手なだけだったかもしれない。今度は山にでも行ってなにが起こるか試してみましょう。

◆初めて手紙を出します。これを書いていて気づいたのですが，「あなたの愛機は」のところに“ない”とあるのにはビックリ。あのAS○○○でもないのに，パソコンも持たずにOh!Xを買っているなんて。そんな人，本当にいるんでしょうか……（いたら失礼)。

佐々木　剛宏(21)北海道

ちゃんと，いますよ。いっぱい。

◆しかし，僕みたいにパソコンも持っていないのにOh!Xを読んでいる人はどのくらいいるんでしょうか。早くX68000を買いたいが，その資金もいつコンポやS-VHSのビデオやレーザーディスクに変わるか定かではない。さらに，スーパーファミコンにもポピュラスやシムシティーが出るという。僕はこの誘惑から身を守り無事X68000を買えるだけのお金をためることができるか。　近藤　英二(19)愛媛県

ほら。

◆もうすぐX68000を買うと思うので，これから“Oh!X”を買っていこうと思っています。面

白い記事をどんどん載せてください。

野坂　隆征(15)滋賀県

ほらほら。

◆最近JR各社は新しい特急列車用車両をどんどん投入していますが，ラップトップパソコンやワープロを使えるように座席の下にコンセントがほしいと思っているのは私だけでしょうか。また，車内の電話からパソコン通信ができるようにデジタル化してほしいと思っているのも私だけでしょうか。　溝渕　誠(19)大阪府

コンセントほしいですよね。ないと洗濯もできないし。ところで，よくビルの外側とかにコンセントがあったりしますけど，あそこにコタツとかをつないで人が住んでしまうんじゃないかと思ってしまうのは私だけでしょうか。

◆ついに明日から北海道だ。待ってろよクッシー，待ってろよロシア兵，待ってろよオーロラの下の白夜の土地よー！（すでになにかを間違えている）俺のVT250Fはもう燃えているぜ！（ちなみにカラーは赤。ううっ，なぜだか恥ずかしいぞ！）ピッカピカに洗車してイオンコートまでしたもんねー！　紅いイナズマと呼んでくれ。さーて，あとは3万円がどれだけもつかだ……（お土産？　ひゅるるるるう〜）。

大津　和之(20)福岡県

3万円だと結構きついような気が。いきなり，いろんなところが壊れて補修費用がかかったりするかもしれないし。やっぱりスナフキンのように歩きで行かないと。それだと，無一文でもなんとかなるし。

◆ああ，2Mバイトユーザーはいいよなって本当に感じてしまった。ワールドコートを見たときにはなんともいえない気持ちになった。それにしても，SPSにひと言いいたい。2Mバイトいるならいると事前にはっきり広告に載せてくれ，もう発売当日にこんな悲しい目に遭いたくない。楽しみにしてたのになあ，ワールドコート（うるうる……）。　今井　武史(16)福岡県

うーん，こういう人は結構いるんだろうな。なんで，2Mバイトいるんだろう。本当に必要ならしょうがないけど。

◆脱「ムー」誌を図った新しい表紙の分析結果を発表します。

4月号　未来都市
5月号　ノアの箱船（かな？）
6月号　UFO
7月号　幽霊の徘徊（こじつけ）
8月号　月の異常接近

うーむ，やっぱり「ムー」してますね！

倉田　泰幸(20)茨城県

ちょっと苦しいけど，よくこじつけましたね。9月号の表紙はどうでしたか。

◆なぜ，ハガキが汚れていると思いますか。な，なんと私の部屋が水浸しになったからなのです。上の階のやつが水を流しっぱなしで放っておいたためだった。天井をつらぬいて落ちてくる水は泥水に変わり果て，2日前に来たばかりのスキャナやディスプレイなどを襲っていた。買ったばかりのOh!Xもグシャグシャであった。これは，外出中に起きたことで防ぎようがなかった。その晩，私は徹夜でリモコンやマウス，スキャナなどをお手入れするはめになった。テスト期間中だった。次の日のテストは全滅だった。部屋はまだ工事中でX68000は友人宅。この怒り，わかってくれます？　野田　博(19)群馬県

本当にハガキが黄色いもんなあ。泣くしかないってやつですね。

◆本日，1990年7月29日（日）午後2時10分，「バック・トゥ・ザ・フューチャー3」を観ました。前2作に負けず劣らず，とても面白かったです。制作スタッフの皆さんには，ただただ敬服の念を抱くばかりです。僕も来年は就職。夢のある職場に就きたいと思っています。編集部の皆さんも暑さに負けず読者に夢を与えてください。　中島　靖(20)福岡県

まだ，パート2も見ていない……。レンタルビデオを借りて見てみようかな。ほかにもいっぱい見たいのがあるんだけれど，なかなか時間がなくて最近全然映画は見ていない。しくしく。

◆対戦ポピュラスの記事，楽しく読ませていただきました。Oh!Xの中では珍しく心底笑わせてくれる記事でした。私も友人のPC-9801相手に勝たせていただいていますが，今回の記事で大変勉強になり，ここ3日12連勝しています。夏休みに入ってから大学に出向く機会があり，ふと電算機室に人がいるのに気づき，先生と話でもしようと行ってみると，電算部のやつらと先生が学校のPC-9801を使ってプロミストランドの対戦をしているのです。考えてみれば学校のものということで費用が一切かからず，いまでは結構なゲーセンと化しています

吉葉　勝幸(19)栃木県

結構強そうですね。ポピュラス大会に応募してくれたらよかったのに。

◆対戦ポピュラスはとても面白かった。祝氏が小差で勝つと思ったら，西川氏が勝っていた。やはり年齢差がものをいったか。マップの作り方もうまいと思う。中野氏も結構強いかもしれない。これからもこのような企画お願いします。

清水　達朗(21)岐阜県

たしかに，中野さんも強いですよ。ひょっとしたら，自分でやるのよりもいじわるなマップを作るほうが好きかもしれないけれど。

◆こんにちは。「対戦ポピュラス」を見て，ついポピュラスを買ってしまいました。9月にはシムシティーが出るというのに困ったなー。ところで，うちのワープロによるとポピュラスを漢字で書くと「歩避ゅ羅州」になるということです。　村上　淳一(19)福岡県

あっ，本当だ。そういうふうに変換される。いつもカナに変換するので，気づかなかったけど。

◆先日，期末試験から解放され，気晴らしに散歩に行った。近くに利根川があるので，その土手に行ってみた。そこで，である。土手を見た瞬間，僕の頭の中をよぎっていったのは「この土手を平地にするには，何回クリックすればよいか」という問題である。思わず考え込む自分が怖くなった。どうやらポピュラスにハマッたようだ。試験勉強もせずにやってたからなあ。だけど，夢にまで出てくるとは……テトリス以来だ。　足立　義宗(15)埼玉県

僕もゲームボーイでテトリスをやりすぎて夢に出てきたことがあるなあ。まあ，どちらもそれほど面白くて熱中できるということでしょうけど。

◆ウルティマVをやってますが，そのあとポピュラスをやって人を沼に落としたり，騎士の周りの土地を削って海で溺れさせたりしていると，自分は本当にアバタールなのだろうかと心配になることがあります（んなわきゃねぇよ！）。あと，ウルティマで誰でもいいからFuck youと入力すると面白いメッセージが返ってきます（徳が崩壊するという噂も……）。

小笠原　洋(16)東京都

そんな下品なことは私にはとてもできな……なるほど。

◆風呂を沸かし始めた。沸くまでポピュラスをやって待つことにした。熱中した。気がつくと風呂のフタが熱でゆがんでいた。翌日の朝，いい湯加減になっていたので，朝風呂に入って学校へ行った。ばちがあたるかもしれん。

▲江副　滋　東京都
いやー，編集部でもワールドコートがなかなか流行っていて，みんなそれぞれ「○○がいい」とかいってますよ。

▲吉田　里志　宮城県
ワンダラーズ・フロム・イースか。あれは最後のボスどころか，途中のザコキャラにやられてしまいあきらめました。

福岡　裕和(21)茨城県

朝風呂はやっぱり気持ちいいよね。

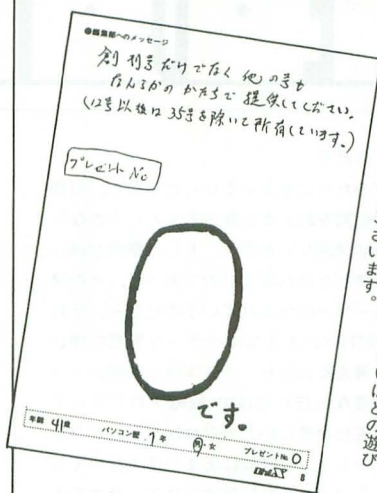
◆祝一平VS西川善司は海原雄山VS山岡士郎を見ているようだった。どうせなら，祝氏に「この祝にOh!Xのスタッフの汚い手垢のついたX68000でポピュラスをやれというのか」とかいって，数日後に満開製作所から新品のX68000が送られてくるとか,「この祝に猿や豚といっしょにポピュラスをやれというのか」とかいってほしかった。残念だ。しかし，完全に「美味しんぼ」に毒されているなあ。

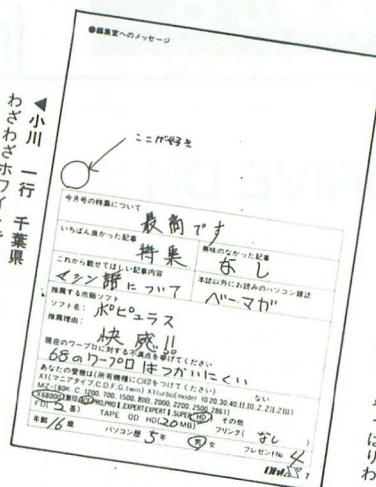
高麗　道也(19)香川県

やっぱり，多かったポピュラスのハガキ。

ということで今月のSTUDIO Xは終わりです。あっ，それと8月号のA.T.さんの編集後記に対する「僕もトマトジュースが好き」っていうハガキもすごく多かったですよ。では，さようなら。

変なハガキ

最近，ちょっと変わったハガキが目についたので特別に載せてみました。

◀小川　一行　千葉県

わざわざホワイトでアンケートハガキの枠を消して書いたんですね。41歳とは思えないほどの遊び心，ありがとうございます。

◀高橋　健司　愛媛県

これはもうさっぱり意味がわかりません。ハガキの裏を見てしまいましたけど。やっぱりわかんなかった。思わず

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集部では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。
●紹介を希望されるサークルは必ず会誌の見本を送ってください。

仲間

★よろずサークル劇団まいむ・まいむでは現在，新規会員を募集しています。現在会員は45名(うち女の子2名)で，主な活動は月1回の会誌発行です。会誌ではゲームレビューやフリートーク，小説やイラストの掲載など，いろいろなことをやっていて，パソコンを持っている人も持っていない人も十分楽しめるようになっています。女の子も入会した劇団まいむ・まいむ，みんなで楽しく活動しましょう。詳しくは62円切手同封のうえ下記まで。入会案内書をお送りします。〒093-05　北海道常呂郡佐呂間町永代176　小野富二雄

★古旗一浩＆EXTRAからのお知らせ。MZ-700/1500，X1/turbo，MSX，etc.のグループ，EXTRAでは会員募集中です。今回はNewサウンドルーチンVer.8.0とMML.Cの配布のお知らせです。割り込みを使用していますので，音楽を聞きながらエディタで修正も可能です。注文の際には，メディア（カセット，QD，FD）を明記してください。詳しくはEXTRAまでご連絡をください。また，サウンドルーチンVer.8.0と上位互換のサウンドルーチンVer.9.0もマニュアルはありませんが専用MML.CIIと配布します。一応Ver.8.0とVer9.0には約100曲楽譜から変換されていますが，現在は配布できません。どの曲が変換されたのかはEXTRA vol.45に掲載予定ですのでそちらを見てください。また，Picture Compilerも新しいバージョン（Ver.1.1）になりましたので，こちらのほうもよろしく。サウンドルーチン以外にもS-OS関係のソフトの配布，未発表のゲームや，System-7Bのグラフィックエディタやツールなども配布しています。また，MZ-700/1500/2500/X1/turbo/MSX/X68000のいずれかでプログラムを組める人も募集中です（特にMZ-700＋System-7C!でプログラムを組んでくれる人)。詳しいことは，皆さんお誘い合わせのうえ，下記の住所までお問い合わせください(パソコンなんて持ってなくてもかまいません)。MZ-700/1500/2500/X1/turbo/MSX/その他のユーザーの皆さん，EXTRAに入会しよう！　〒811-42　福岡県遠賀郡岡垣町戸切794-3　筑紫高宏

★サークル「夢幻史」では第2期後期会員募集中です。X1/turbo，PC-8801ユーザーを対象としており，活動内容は2カ月ごとに発行する30ページ前後の会誌が中心。BASIC，音楽，雑談，Q＆Aなどの記事を掲載。あと，自作プログラムの発表なども。また，会誌の原稿を書いてくれるスタッフも募集しています。興味のある方は62円切手同封のうえ下記まで。〒593　大阪府堺市東八田40-7　藤田淳一「Oh!X読んだぜ」係

売ります

★コルグのT3（新同）をマニュアル，保証書，付属品一式すべて付けて専用スタンド「KSSB」といっしょで27万円で譲ります。MIDIケーブル（5m）を2本サービスします。連絡はハガキか手紙でお願いします。〒328　栃木県栃木市惣社町1619　毛塚健次(17)

★オムロンのモデム「MD24FP」とジャストシステムのハンディカプラ「JS-HC001」のセットを送料先方払いで5万円。新品同様，付属品，マニュアル，保証書すべてあり。連絡は往復ハガキで。〒713　岡山県倉敷市玉島中央町3-1-39　松浦正章(18)

★カラーイメージユニット「CZ-6VT1」を3万円程度で。新品同様。連絡はハガキでお願いします。〒629-23　京都府与謝郡野田川町岩屋356　安田真二(17)

★X68000用数値演算プロセッサボード「CZ-6BP1」を2万5千円（送料込み）で。完動，箱，マニュアル，付属品すべてあり。連絡は往復ハガキで。〒960-02　福島県福島市飯坂町平野字八龍前1-20　大内一之(18)

買います

★X68000用増設RAM「CZ-6BE1」を送料込みで2万円ぐらいで（完動，付属品付き)。連絡はハガキで。〒341　埼玉県三郷市彦成3三郷団地9-9-1203　白木準三(38)

★X1用データレコーダ「CZ-8RL1」(完動，箱，マニュアルあり)。送料込み4千円から6千円で。〒355　埼玉県東松山市材木町6-14　杉田裕之(17)

バックナンバー

★Oh!Xの1988年12月号，1989年6,7,8月号を送料込み各1200円で。切り抜き不可。傷，汚れは可。連絡はハガキで。〒515　三重県松阪市上川町421　松浦邦博(20)

★Cマガジンの1989年10月号から12月号を。切り抜き，汚損不可。1冊2千円で。連絡は往復ハガキで。〒311-04　茨城県日立市東河内町1227　石川満章(22)

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今回は8月号の内容に関するレポートです。

●「ADVANCED 2D GRAPHICS」について。いきなり難解な数学の話になるのだろうかと思ったらグラフィックツールの紹介から入っているのは初心者でも安心して読めたと思います。そして，やはりきましたね，Oh!Xならではのストロングな記事が，アンチエリアスという言葉は日頃グラフィック画面をただのグラフ表示か，たまにやってもレイトレ程度という私にはよくわかりませんでしたし，恥ずかしながらジャギーという言葉は初めてです。要するにペイントするときの境界処理ですね……。とにかく，理解するためにもペイント関数を早速「SLANG」に移植すべく解析してみます。ソドリッドスキャンコンバージョンについては以前「MAGIC」（いまは読者のうち「MAGIC」を知っているのはどれくらいかなあ）掲載のときに同じような解説が多角形表現についてなされていたので予備知識もありわかりやすかったです。ジャギーの除去もなんとか理解できましたが，それ以降は私にはよく理解できませんでした。でも，プログラムを解析してでもモノにしたいですね。Oh!X（MZ）はドラゴンだ（懐かしい！）というポリシーが表れた特集だと思います。

梅本英之(20)　X1Gmodel30，PC-8201，PC-1251　奈良県

●バグのために暴走するならともかく，引数が制御範囲を超えると暴走するというのならば，一切問題ないと思う。もし，暴走が怖いのなら使わなければよいのであって，その選択はユーザーに任されているのだから。それに，XROT0.Xのようなルーチンを実際に使いたいと考える人なら，バスエラー回避のルーチン程度なら作れるはず（回転プログラムで画像の劣化が激しいともなればゲームとプレゼンテーション以外にはあまり使えないでしょうから）だし，それ以外の人はどうせオモチャにしかならないのだから，無視してもかまわないだろう。「その筋」なプログラムの登場を歓迎をします。

高村信(20)　X1turbo model30，PC-8001mkII　東京都

●正直いって「まだ100号か」という気がします。僕が買い始めたのは1984年からですが，それよりもずっと前からOh!X（MZ）はあったかのような錯覚に陥るのです。

しかし，それとは逆に僕の本棚に収められた計83冊のOh!MZやOh!Xを見ていると6年という月日がとても短く感じられるのです。僕にとってOh!MZからOh!Xへ変わったのはほんのちょっと前のことであり，X68000の登場もつい最近のことのように思えます。それは，Oh!MZ，Oh!Xともに持っている情報の普遍性から来ることではないでしょうか。

たとえば，僕はいまだに他機種からのBASICプログラムの移植には1986年1月号からの対照表を使うし，プリンタについて凝りたくなればプリンタ関係特集をやっているOh!X（MZ）を全部出してきます。

つまり，Oh!X（MZ）の記事はほとんどが（すべてとはいいません）タイムリーな情報だけを扱ったものではないのです。そこがポイントなのです。そこがパソコン情報誌の本来負うべきところであるのです。このような雑誌はほかの分野に類を見ないでしょう。何年たっても使える雑誌など考えられません。だから，これからもOh!Xのスタッフは変わっていくでしょうが，Oh!MZの持っていた素敵なパソコンの使い方をこれからも載せていってほしいと思います。

長谷川敦士(17)　MZ-2521，MSX2　山形県

●私はこのCARD.FNCがここまでメジャーになることを予想できませんでした。それだけにこのCARD.FNCがよくできていたということ，ひいてはユーザー側で関数をメーカーのものとまったく同様に拡張できるといった優れた汎用性（拡張性）があったということでしょう。こういうときに「X68000でよかったなあ」と思うわけです。ついでに，私個人が希望したいものとしては，RS-232Cを使っての対戦型ポーカー，GP-IBを使って10人プレイ可能な大富豪，SCSIインタフェイスを使ってのスピードなんかがやりたいですね（はっきりいってむちゃですね）。

浅野憲(19)　X68000PRO，X1turbo III，X1Fmodel20，MZ-80C，PC-6001，M5Jr.，PC-1245　大阪府

ごめんなさいのコーナー

9月号　BILLIARDS

リストに一部見にくい箇所がありました。大変申し訳ありませんでした。以下に正しいリストを掲載いたします。

P.148　リスト2

18 var a,b,bk,……,l,rf2,v

9月号　ハンディイメージスキャナアダプタの製作

P.91の図3の回路図およびP.92の表1の部品表に一部誤りがありました。10kΩの抵抗が1本抜けています。図1が正しい回路図です。お詫びいたします。

また，図2のようにするとカラースキャナが使えます。電源は12Vのものを使用します。ただし，取り込みプログラムは各自で作成してください。

図1

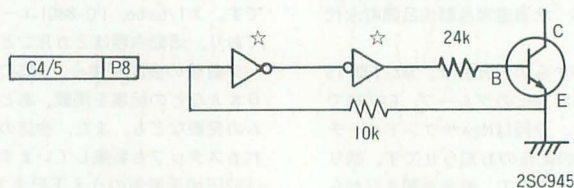
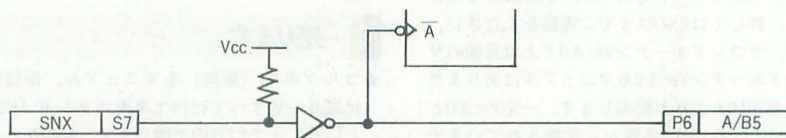



図2

バグに関するお問い合わせは
☎03(5488)1311(直通)
月〜金曜日16：00〜18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。

また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

はっきりいって オーバーフローだ もっとページを！

▼X68000の2大特殊機能といえば，なんといってもサウンド＆グラフィック。ですが，記事の反響は，目に見えるだけにグラフィックのほうがいいのが普通です。プログラム自体が難解なグラフィックに比べ，サウンドのほうはデータの作り方さえわかれば比較的手軽に本体の機能を引き出せます。音楽はわからないなどといわず，織毛氏の用意したサンプルなどを入力して聞いてみてください。

▼以前予告した，同人ソフトの紹介ページが今月の「THE USER'S WORKS」です。今後もできる限りカラーページを使って紹介していきたいと思いますので，ぜひとも編集部までお送りください。お待ちしております。

▼THE SOFTOUCHではAFTER REVIEWに載せる皆さんのゲーム評価を募集しています。記事には載っていなかったけど実はここが面白かったとか，なんとかシステムはこうすればもっとよくなるとか，やり込んだ人だけにこの楽しさがわかる……など，実際にプレイした感想をハガキのメッセージ欄に記入してお送りください。ちょっと古いゲームでもかまいません。採用させていただいた方には記念品をお送りいたします。

▼というわけで今回は，Oh!MZの創刊3号（1982年8月号)を3名の方に差し上げたいと思います。ご希望の方は綴じ込みのアンケートハガキのプレゼントNo.に0と記入してお送りください。

そして，お待ちかね「創刊号プレゼント」の当選者発表です。森　敦史さん（東京都)，平義晴さん（大阪府)，赤城豊和さん（兵庫県)の3名の方が当選されました。おめでとうございます。また，STUDIO Xに掲載された小川一行さんにも特別に創刊号をお送りしたいと思います。

▼今月はページの都合でいくつかの連載がお休みになってしまいました。「マシン語カクテル in Z80's Bar」「X-BASICプログラミング調理実習」「PASCALプログラミングへの招待」は次号までお待ちください。

投稿応募要領

- 原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。
- プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，掲載にあたっては，編集上の都合により加筆修正させていただくことがありますのでご了承ください。
- ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討のうえ，製作したハードが必要な場合はご連絡いたします。
- 投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒108 東京都港区高輪2-19-13 NS高輪ビル
ソフトバンク出版部
Oh!X「㋢㋮㋯名」係

SHIFT・BREAK

▶今月は怒濤のFM音源特集。ちょっと難しいけど一度は楽しんでほしい世界だあね。なんといっても人に自分のプログラムを聞かせるのが醍醐味だから。悪友に聞かせて「けちょんけちょん」に言ってもらうとかなり上達するからやってみるべし。井の中の蛙が飛び出して蛇にのまれて胃の中の蛙になったんじゃ洒落にもならんしの（意味不明)。（織）

▶オーディオぐらい初見でいじれるし，パーソナルワープロだって打てる。こんなところにいるぐらいだから電化製品に弱いつもりは全然ないのだが……。エラー音が出るたびにびびりまくるのは多機能電話。再生押して，短縮押しながら3押して……と，操作は隠しコマンド以上である。誰か助けてぇとOA恐怖おじさんの身になっている昨今だ。（浦）

▶何を勘違いしたのか，Communication PRO-68KのVer.1.0を買ってしまった。しかし，それにもめげずパソコン通信にハマルのであった。世の中のフリーウェア＆PDSはよくできていて，投稿すればずいぶん原稿料がもらえるのになあと思うものも多い。それを無料で配ってしまうとは，なんとお心の広い。感謝，感謝。（亀）

▶最近，頭の中身がポエムである。または，鳥頭とでもいうべきであろうか。1週間も前にあったことなど，ほとんど忘れているし，同窓会の連絡が回ってきたときには，30分待ち合わせ時間を聞き間違えた。同窓会では2/3ほど名前のわからない人がいて，家に帰ったあと，卒業アルバムを見ながら復習をしてしまう僕であった。（純）

▶某店でアサルトのROMを1,000円で打っていた。これは「買い！」と思い店員に声をかけるとパネルがないのですが……」との答え。そうかアサルトは特殊パネル（あの2つのジョイスティック上にボタンのついてるヤツ）だったのだ。だれか，アサルトの特殊パネルを譲ってください！

（システムIIがあれば彼女はいらないと思う善）

▶こつこつためる。ちょっとたまったところでいい気になってハデに使う。なぜか直後に大きなマイナスがくる。じり貧になる。一発逆転を狙って思い切った行動に出る。しくじる……。教訓：余裕を持って使いたいものですね，お金もマナも。ポピュラスは人生の縮図だったのだ。それでは対戦ポピュラスを人生にたとえてみよう。きっと怖いよ。（A. T.）

▶なかんずく，私にも落ち込むときはある。たとえば，自分が書きたいと思っていた内容を，自分よりずっとうまく，面白く書かれたときである。才能のなさを感じる。続いて，自分の写真をあるべき場所ではないところで見たときである。生理的に恥ずかしいものがある。いうまでもなく，可愛い女の子に冷たくされたときも落ち込む。一応ね。（K）

▶結局，駐車場が見つからないまま車の契約をしてしまった。しょうがないので不動産屋に行ってみたが，不動産屋は駐車場だけを探しに来る客に対しては愛想が悪い。そこを我慢して話をしたのだがやはり空いた駐車場はない。納車は駐車場を見つけて2週間後とディーラーに言われているが，果たしてそれはいつだろう。（どうにかなるさのKO）

▶「MSXturboR」の発表会に半分仕事，半分趣味で行ってきました。Z80には思い入れがあるのでワクワクしてたんだけど，実際の仕様を見てみると，AV機能も強化という割には，MSX2＋の CPU をR800にしてPCM音源を積んだだけみたいなんですよね。ちょっと期待外れかな。R800だけでも心を惹かれるんですけど，それだけじゃあね。やっぱり。(A)

▶ああ，夏が終わった……。思えば今年の夏は引っ越しに始まり引っ越しに終わったんだよなぁ。サーファーだった私が1回しか海に行かなかったなんて……。まあいい，副編集長も誕生したことだし，冬はたっぷりスキーに行かせてもらおうっと。さぁて，新しいウェアでも買いにいこうかな。

（薄い水着の跡が悲しいE. O.）

▶帰省してみると軒並みX68000ユーザーが増えていることに驚いた。さて，KO氏に刺激されたのか，某所の社長さんがいきなりベンツの値段を聞いてきた。うーむ，この人はいったい……。でもどうやら本気で車（それも外車）を買う気になったらしい。そこで，とりあえず強力にボルボをすすめておいた。さて？（U）

▶FMの音源の特集なわけだが，X68000ユーザーでも肝心なOPMドライバを持っていない人が僅かながらいる。初代機で，かつ，ほとんどシステム環境を整備していない人だ。いま持ってるソフトの中身はチェックしておくといい（なにか買えばたいてい付いてくる)。とりあえず，SX-WINDOWを買うのがいちばんお得かな。IOCS. Xも入ってるしね。（T）

microOdyssey

東京ってやつは不思議な街だ。来る者を拒まないこの街は，なにかとてつもないパワーと引力を持っているかのように君臨している。誰もがこの街にあこがれ，集まってくる。かくいう私も，生まれてから25年間この街を離れらずにいた。けど，なにかがいつも足りないのだ。こんなになんでも揃っていて，飽きさせないだけのオモチャや場所も腐るほどあるのに，なにか，足りない。刺激に慣れちゃったのかな，なんてナマイキにも思ってたりしたんだけど……。

でも。独り暮らしを始めてしばらくしたときに，わかったね，足りないものがなんなのか。いいきっちゃうよ，もう。足りないもの，それは人間同士のコミュニケーションなのだ。

いまの日本は雑誌やTVが氾濫しきっている。いわゆるマスメディア全盛ってやつだ。我々はそれらに取り囲まれているというよりは，逆にそれらに翻弄させられている感がある。雑誌にオシャレな店が載っていると聞けばそこへ走り，新聞のテレビ欄や電車の中吊を見れば，まるでその番組や記事を熟知したかのように錯覚させてくれる(ように書かれている)。もちろん流行もすべて情報まかせ。で，マスメディアってやつは主に東京を中心にはびこっている。そのド真ん中に住んでいるもんだから，安心感も手伝って余計に自分からなんかしようなんて思わなくなるし，すべてが受け身，それで十分周りに順応できるように仕組まれている。特別他人と話をする必要，すなわち他人とのコミュニケーションを持たなくても平気なように，だ。でも，これってかなり非人間的なことよねぇ。

なぜ，こんな寂しいことに気がついてしまったのかというと，私は独り暮らしを始めて6年間同じアパートにいたにもかかわらず，他の7人の住人と一切話をしたことがなかったからだ！　6年といえば，中学と高校を卒業できてしまうぐらいの期間だぞ，おい。その間〝同じ屋根の下〟にいた人たちとすれ違ったのだって20回くらい，はっきりいってこれは異常！

確かにこの街はいつも動いているわけだから，そんななかで生活するのはかなり疲れる。だから，わずらわしいコトはどっかに置いといて，情報を仕入れてラクに暮らしたい。が，それが悪循環の始まりだったりする。疲れるからと周りとのコミュニケーションを断つ。でも，ひとりでいるのは寂しい。友達はへっちゃったし……。さて，どうしよう。そこでいきなり異性にいくんだな，これが。で，同棲，2人の世界よ。私のいたアパートなんてほとんどそうだったもん。女の子しか住んでいないはずのアパートのそこかしこに男の声が聞こえる，げげっ。コミュニケーションの少なさ＝世間の狭さだよね。いいよ，同棲したって。でもね，あとどうすんの，振られたとき。もともと寂しくて始めちゃったもんだから，いなくなったらもうタイヘン。なぐさめてくれる友達もすでにいなくて，さ。

いま，横浜に引っ越してよかったと思っている。団地のせいもあるけど，全然知らない人なのにちゃんと挨拶してくれる。それがすごくうれしい。あったかい気持ちになるもん。人間同士のコミュニケーション，やっぱりそれって人を好きになるうえでの最低条件だよね。　(E. O.)

1990年11月号10月18日(木)発売

特集　理科系のGAME REVIEW
続報 C compiler PRO-68K
ドローツール CANVAS PRO-68K
全機種共通システム
スクリーンエディタ
CARD. FNC用ゲーム
ローリングストーン

バックナンバー常備店

東京	神保町	三省堂神田本店5F 03(233)3312
	〃	書泉ブックマートBI 03(294)0011
	〃	書泉グランデ5F 03(295)0011
	秋葉原	T-ZONE 7Fブックゾーン 03(257)2660
	八重洲	八重洲ブックセンター3F 03(281)1811
	新宿	紀伊国屋書店本店 03(354)0131
	高田馬場	未来堂書店 03(200)9185
	渋谷	大盛堂書店 03(463)0511
	池袋	リブロ池袋店 03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム 03(981)0111
神奈川	横浜	有隣堂横浜駅西口店 045(311)6265
	〃	有隣堂ルミネ店 045(453)0811
	藤沢	有隣堂藤沢店 0466(26)1411
神奈川	厚木	有隣堂厚木店 0462(23)4111
	平塚	文教堂四の宮店 0463(54)2880
千葉	柏	新星堂カルチェ5 0471(64)8551
	船橋	リブロ船橋店 0474(25)0111
	〃	芳林堂書店津田沼店 0474(78)3737
	千葉	多田屋千葉セントラルプラザ店 0472(24)1333
埼玉	川越	黒田書店 0492(25)3138
	川口	岩渕書店 0482(52)2190
茨城	水戸	川又書店駅前店 0292(31)0102
大阪	北区	旭屋書店本店 06(313)1191
	都島区	駸々堂京橋店 06(353)2413
京都	中京区	オーム社書店 075(221)0280
愛知	名古屋	三省堂名古屋店 052(562)0077
	〃	パソコンΣ上前津店 052(251)8334
	刈谷	三洋堂書店刈谷店 0566(24)1134
長野	飯田	平安堂飯田店 0265(24)4545
北海道	室蘭	室蘭工業大学生協 0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は綴じ込みの振替用紙の「申込書」欄にある「新規」「継続」のいずれかに○をつけ，必要事項を明記のうえ，郵便局で購読料をお振り込みください。その際渡される半券は領収書になっていますので，大切に保管してください。なお，すでに定期購読をご利用の方には期限終了の少し前にご通知いたします。継続希望の方は，上記と同じ要領でお申し込みください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

10月号
■1990年10月1日発行　定価560円(本体544円)
■発行人　孫　正義
■編集人　橋本五郎
■発売元　ソフトバンク株式会社
■出版事業部　〒108 東京都港区高輪2-19-13　NS高輪ビル
Oh!X編集部　☎03(5488)1309
出版営業部　☎03(5488)1360 FAX 03(5488)1364
広告センター　☎03(297)0181
■印　刷　凸版印刷株式会社

満開の電子(でんこ)ちゃん

作：いわいいっぺい
え：岡村祭

西川善司（埼玉県）

よく昔から「隣の客はよくきゃき食う柿だ」などと言いますが、私はスイカが大好きです。そんな時「なだしお」の如く現れたのが電脳倶楽部でした。届いたディスクの内容の多彩さにびっくり。しかし、何より嬉しかったのはプログラムの殆どにソースリストがついていることでした。X68Kをバリバリに使っている人達のソースを月千円で見れてしまうとは、私はなんてラッキーボーイなんだろうと神様に感謝してしまったくらいです。特に通信をやっていないユーザーには必携のアイテムですね。

HOST PRO-68K シリーズに新製品登場.!!

HOST PRO-68K Personal SS-502

1回線専用ホスト入門用ソフト

HOST PRO68K Personal 概要

対応回数	1回線
使用モデム	ATモデム MNP(RTS/CTS)可
通信速度	最大9600bps
会員数	最大999人
掲示板数	最大40個
機能	電子掲示板・電子手紙・電子会議(チャット)・会員情報

これらは、コンフィグファイルで設定できます。

注1:このホストはテキスト形式の転送方法を採用しております。

特長

- 各種設定のコンフィグファイル化
- RS−232C回線とは別にキーボードからのアクセス、ダウンロード、アップロードが可能
- モニタで、各チャンネルのユーザーの打ち込んだコマンドや通信状態を確認
- 各掲示板別にSIG、ボードパスの設定
- メンテナンス作業がオンライン実行可能(ボードインデックス、メールインデックス)
- オンラインサインアップ等、ゲストへの設定が可能
- 行編集(オンライン簡易エディタ)機能
- その他
  シスオペレベルで会員情報の変更が可能
  タイムアウトによる回線切断
  PDS専用掲示板の採用
  (1書込中で、ドキュメントとテキストプログラムの分離)接続MNPタイプの識別
  ログイン、ログアウト時間の記録
  非アクセス時のモニタ画面消去可能

<差額交換について>

回線を多回線に拡張する場合ユーザー登録を行っていただいた方に限り多回線用ホストと差額にて交換をさせていただきます。

HOST Personal ➡ HOST 3 ¥20,000
HOST Personal ➡ HOST 9 ¥40,000

好評発売中

HOST 9 PRO-68K 9回線用 ¥59,800円

HOST 3 PRO-68K 3回線用 ¥39,800円

SPS-NET
TSUKUMO-NET モデル運用中!!

たーみのる 2

今、X68000の通信が変わる!!!

ユーザー重視の機能を搭載して
好評発売中
17,800円

24/31KHz ディスプレイ対応

「た〜みのる」が装いも新たに「た〜みのる2」として登場!
「た〜みのる」が通信入門版なら「た〜みのる2」はマニアタイプの通信ソフトです!!!

X68000専用
パソコン通信ソフト

「た〜みのる2」はX68000用に製作された通信ソフトです。X68000の機能を充分に引き出して、ユーザーの方々が簡単に操作できるよう工夫・製作されています。

SPS-NET

ホストコンピュータ
X68KX2台
(HOSTPRO68K16)使用

TEL (0245)46-1167(代)
Tri-P接続ホスト局

通信制御手順

漢字・文字コード:シフトJIS
通信速度:300/1200/2400(自動判別MNP7)
通信方式:全二重
データビット長:8ビット
パリティー・チェック:なし(NONE)
ストップ・ビット:1ビット
フロー制御(Xコントロール):行なう(XON)
シフト制御(Sコントロール):行なわない(SOFF)
ゲストID:GUEST
ゲスト・パスワード:なし

入会方法

登録料¥3,000(税別)
会費無料

下記の用紙に直接記入するか又は、コピーして記入し、72円切手同封の上、「SPS-NET係」までお送り下さい。届き次第、仮登録を行いID発行後SPS-NET専用の郵便振込み用紙ならびに運用の手引きをお送りいたします。それに従い、3ヶ月以内に登録料3,000円(税別)を御入金下さい。
入金確認後正式会員として再登録します。

例◎パスワード=SPS-NET (8文字まで大小文字の識別あり)
( )

◎本名=大和大五郎(8文字まで)
( )

◎ペンネーム=大ちゃん(4文字まで)
( )

◎年齢=30(現在の年齢)
( )

◎電話=0245-45-5777(市外局番から)
( )

◎職業=株式会社エス・ピー・エス(16文字まで)
( )

◎住所=福島市太平寺字町ノ内5-3(24文字まで)
(〒)

◎自己紹介=SPS-NETをよろしく (24文字まで)
( )

◎システム構成=X68000ACE-HD MD2400B (18文字まで)
( )

プログラマ募集!!

SPSでゲームを作ってみませんか?

アセンブラでプログラムの組める優秀な人材を若干名募集しています。就職希望の方は62円切手同封の上、「就職案内係 大和」までお手紙ください。折り返し就職のご案内をお送り致します。
尚、デザイナー、音楽プログラム等の専門職は募集しておりません。

〒960 福島市太平寺字町ノ内5-3 ☎(0245)45-5777
FAX(0245)45-1804(GII,GIII)

当社の製品は全国の有名デパート、パソコンショップでお求めになれます。尚、お求めになれない場合、郵便局にてお申し込みください。●口座番号 郡山5-12298 ●加入者名㈱エス・ピー・エス ●金額 代金に3%の消費税を加算した額 ●通信欄(裏面)ご希望ゲームソフト名、数量、代金合計、年齢、氏名、機種名、テープかディスクの種類。(一週間以上かかりますので、お急ぎの方は現金書留をご利用ください。その場合、おつりのいらないようにお願いします。

■表示価格に消費税は含まれておりません。

X

micro computer shop

BASIC HOUSE

宇都宮店&大田原店

PRO SHOP

CZ-653C……¥285,000
CZ-603D……¥ 84,800
定価合計……¥369,800
Basic House特価

Super HD

CZ-623CTN……¥498,000
CZ-613DTN……¥135,000
定価合計…………¥633,000
Basic House特価

ExpertII

CZ-603C………¥338,000
CZ-613D………¥135,000
定価合計…………¥473,000
Basic House特価

台数限定 電子手帳特別セット

A SET	B SET
PA-8600	PA-8600
CE-200L	CE-200L
CYBERNOTE PRO68k	Stationary PRO68k
Basic特価 ¥41,800	Basic特価 ¥37,800

PLAY THE MIDI MUSIC

A SET	B SET	C SET
CM-32L	CM-64	MT-32
SX-68M	SX-68M	SX-68M
MusicStudio Mu-1	MusicStudio Mu-1	MusicStudio Mu-1
Basic特価 ¥93,000	Basic特価 ¥143,000	Basic特価 ¥88,000

X68000用ハードディスク

アイテックITX-680……¥198,000
アイテックITX-640……¥158,000
アイテム　HXD-040……¥118,000
アイテム　HXD-042……¥128,000
ロジテックSHD-40……Basic特価
シャープ　CZ-64H……Basic特価

X68000用SCSI予約大募集

シャープ
光磁気ディスクユニット
CZ-6MO1　予約受付中
SCSIボード
CZ-6BS1　予約受付中

X68000用PRINTER'S

シャープCZ-8PC4……¥ 99,800
シャープCZ-8PG1……¥130,000
シャープCZ-8PG2……¥160,000
シャープCZ-PK10……¥ 99,800
エプソンAP-850……¥ 94,800
エプソンAP-550EX……¥ 62,800
エプソンVP-1350……¥ 94,800
エプソンVP-2050……Basic特価
NEC　PC-PR201GS……Basic特価
スターCR-3415CL……Basic特価

通信関連品

NEC COMSTER2424/4……Basic特価
NEC COMSTER2424/5……Basic特価
OMRON MD24FS5……¥ 49,800
OMRON MD24FS7……¥ 64,800
CommunicationProV2……¥ 19,800
た～みのる2……¥ 17,800

ゲーマー必須アイテム

CYBER STICK……¥ 23,800
XE1-AP……¥ 13,800
XE1-PRO……¥ 9,800
XE1-ST……¥ 4,900

グラフィックツール

スキャナパラレルボード……¥ 29,800
CZ-8NS1……¥ 19,800
GT-6000……Basic特価
GT-1000……¥ 79,800
HS-10RII……¥ 49,800
HS-7RII……¥ 39,800
CZ-6VP1……¥198,000
IO-735X……¥248,000
ジーズスタッフPRO68K……Basic特価
デジタルクラフト……¥ 39,800
マジックパレット……Basic特価
サイクロンExpress α68……¥ 98,000

その他周辺機器

拡張I/O BOX……¥ 88,000
アンプ内蔵スピーカー……¥ 36,600
カラーイメージユニット……¥ 69,800
ビデオボード……¥ 21,000
RGBシステムチューナー……¥ 33,100
CRTフィルター……¥ 19,800

CRT

CU-21HD……¥148,000
CZ-605D……¥115,000
CZ-604D……¥ 94,800
CZ-613D……¥138,000
CZ-603D……¥ 84,800

NEWS 1

ロケットキャッシャー完成！

新アルゴリズムの採用により従来比約3倍の高速化を実現。ハードディスクキャッシャーのみのバージョンアップとなります。旧製品のディスクのラベルを同封のうえ、1,500円（送料・税込）を現金書留でお送り下さい。

NEWS 2

ビデオボード（CZ-6BV1）を外付けに！

ビデオボード収納ケース（KGB-BVBX）

近日発売予定

（通信販売のみで一般販売はしません）

全国どこでも発送可　長期クレジットＯＫ　送料全国均一¥1,000　宅配便にて即日配送

株式会社計測技研
本社営業部／マイコンショップ／通販部　宇都宮市竹林町503—1　TEL0286-22-9811　FAX0286-25-3970
大田原営業所／マイコンショップ　大田原市美原1—13—4　TEL0287-23-5352　FAX0286-23-5364

マイコンショップ BASIC HOUSE　お申し込み・お問い合せは ☎0286-22-9811（代）

2枚のボードが1枚になった

KGB-X68PRK

※写真はKGB-X68PRK-14です

広大なメモリ空間を実現する最大4Mバイトの

高速増設メモリ

高速演算を約束してくれる

数値演算プロセッサ

- メモリアクセスノーウェイトによる高速アクセス
- CZ-6BE2/CZ-6BE4/CZ-6BP1との混在が可能です
- 複数枚のKGB-X68PRKの実装が可能です
- ジャンパの変更により任意のアドレス空間にメモリの配置が可能です
- ジャンパの変更により数値演算プロセッサの1枚目2枚目/未使用の選択が可能です
- 1M/2M/3Mメモリモデルは購入後にメモリをボード上に追加可能です
- 数値演算プロセッサにはデバイスドライバ(FLOAT3X)が付属します

※メインメモリが1Mバイトの機種では2Mバイト以上に拡張してある必要があります。
※メモリアクセスノーウエイトのため拡張I/Oボックスでは動作しません。
※メモリの増設は当社にボードを送り返していただいて行ないます。

製品価格一覧

製品	価格
KGB-X68PRK-01 (1Mメモリ/数値演算プロセッサ無し)	¥ 58,000
KGB-X68PRK-02 (2Mメモリ/数値演算プロセッサ無し)	¥ 74,000
KGB-X68PRK-03 (3Mメモリ/数値演算プロセッサ無し)	¥ 98,000
KGB-X68PRK-04 (4Mメモリ/数値演算プロセッサ無し)	¥122,000
KGB-X68PRK-10 (メモリ無し/数値演算プロセッサ付属)	¥ 72,000
KGB-X68PRK-11 (1Mメモリ/数値演算プロセッサ付き)	¥ 96,000
KGB-X68PRK-12 (2Mメモリ/数値演算プロセッサ付き)	¥112,000
KGB-X68PRK-13 (3Mメモリ/数値演算プロセッサ付き)	¥136,000
KGB-X68PRK-14 (4Mメモリ/数値演算プロセッサ付き)	¥160,000

購入後の増設費用

メモリ	
1Mバイト	¥24,000
2Mバイト	¥51,000
3Mバイト	¥76,000
数値演算プロセッサ MC68881RC16	¥38,000

充実のBASIC HOUSEハードウェア&ソフトウェア

製品	価格
高速12BIT, 16CH A/Dコンバータボード(KGB-AD12) X1	¥118,000
フォトアイソレーション16BITデジタル入出力ボード(KGB-PIO) X1	¥ 42,000
ハードディスクインターフェースボード(KGB-HDIF) X1	¥ 16,000
アイソレーション16BITデジタル入出力ボード(KGB-X68PIO)X68000	¥ 68,000
ハンディプリンタ&インターフェース(HANDYPRINTjack)X68000	¥ 24,800
高速12BIT, 4CH D/Aコンバータボード(KGB-DA4) X1	¥ 98,000
汎用ローコストA/D&PIOボード(KGB-X1S) X1	¥ 19,800
高速12BIT, 16CH A/Dコンバータ(KGB-X68ADC) X68000	¥128,000
64180CPUボードMach180(KGB-CPXB) X68000	¥ 98,000
ローコストMIDIインターフェース(MELODY BOX) X68000	¥ 16,800

製品	価格
BASIC拡張関数パッケージ(B6-6301)	¥9,800
C言語ライブラリ(B6-6305)	¥6,800
ディスクキャッシャー(B6-6304)	¥6,800
Toys & Tools (B6-6307)	¥6,800
BASIC拡張関数パッケージC言語ライブラリ付(B6-6306)	¥ 14,800
アイコンエディタ(B6 -6303)	¥4,800
CP/M68Kエミュレータ(B6-6302)	¥ 19,800

BASICHOUSE TECOSYS NET
TEL 0286-27-1829 1200 2400ボーMNPクラス5 8ビット/パリティ無し X制御無し ゲストIDなし(オンラインサインアップ)

全国どこでも発送可 長期クレジットOK 送料全国均一¥1,000 宅配便にて即日配送

株式会社計測技研
本社営業部 マイコンショップ 通販部 宇都宮市竹林町503-1 TEL0286 22 9811 FAX0286 25 3970
大田原営業所 マイコンショップ 大田原市美原1-13-4 TEL0287 23 5352 FAX0286 23 5364
マイコンショップ BASIC HOUSE お申し込み・お問い合せは ☎0286-22-9811(代)

新店舗オープン記念セール実施中！

（京王線・北野駅前）

★お買い上げの方に記念品贈呈！

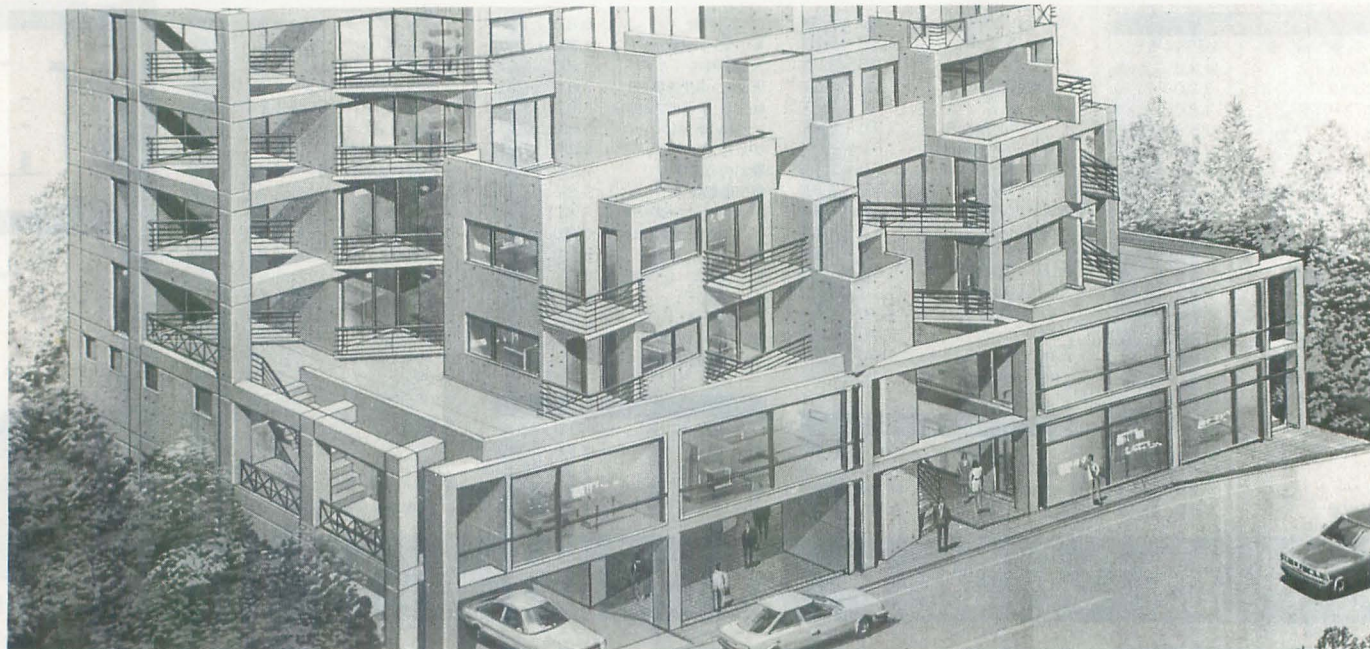
AIBIT
アイビット電子株式会社

X68000 セット

'90年10月末迄

CZ-653C + CZ-602D	CZ-602C + CZ-603D	CZ-612C + CZ-603D
記念特価¥275,000	記念特価¥270,000	記念特価¥358,000

9月30日まで期間中に限り、大奉仕特価品！

CZ-8PC4（80桁ネッテン、カラー漢字）　特価¥64,800
CZ-8PK10（130桁ドット漢字）　特価¥69,000
CZ-8NJ2（サイバースティック）+CZ-239AS（サンダーブレード）
定価¥33,300 → 特価¥23,800

アイビット推奨ディスプレイ

●シャープCZ-860D・BK
カラーディスプレイ
0.31チルト付A/D 15/24
定価¥92,200⇨
特価¥59,800

CZ-860D対応パソコン機種：CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。NEC PC-8801・9801シリーズ（XA・XLのみ不可）MZ700/1500/2000/2200/2500各シリーズ（推奨品シャープ8D8K）。（ドットヒッチ0.39）

●シャープ CZ-603D-GY・BK
（15型カラーディスプレイ）
ドットピッチ3.9
定価¥84,800⇨
特価

CZ-603D対応パソコン機種：※X1シリーズ/※X1 turboシリーズ/X1 turbo Zシリーズ/X68000シリーズ/PC8801シリーズ/PC-9801シリーズ/PC-286シリーズ
（※は接続ケーブルAN1506が必要です）

●シャープCZ-830D・BK
（14型）
2モードオートスキャン方式
（アナログ/デジタル）
定価¥98,000⇨
特価¥54,800《在庫限り》

CZ-830D対応パソコン機種：CZ880C/881C。X1/TURBOシリーズ。ケーブルは本体付属を使用。NEC PC-8801・9801シリーズ（XA・XLのみ不可）MZ700/1500/2000/2200/2500各シリーズ（推奨品シャープ8D8K）。

●シャープCZ-602D-BK
（15型アナログTV/3モードオートスキャン）
定価¥99,800⇨
特価¥75,000

CZ-602D対応パソコン機種：※X1シリーズ/※X1 turboシリーズ/X1 turboZシリーズ/X68000シリーズ/PC8801シリーズ/PC-9801シリーズ/PC-286シリーズ
（※は接続ケーブルAN1506が必要です）

●三菱XC-1498CII
（14型アナログ）
ドットピッチ0.28
定価¥107,000⇨
特価¥59,800

XC-1498CII対応パソコン機種：PC-9801シリーズ/PC-286シリーズ/PC-386シリーズ/PC-8801シリーズ
（上記機種には付属の接続ケーブルで、接続可能）

※シャープ周辺機器（拡張、プリンター他）も常時取り扱っております。

SHARP AX286L-F ラップトップ
定価¥428,000
➡特価¥238,000

SHARP AX286N-H2 All In Note
定価¥398,000
➡大特価！

NEC PC-9801n NOTE
定価¥248,000
➡特価¥198,000

TOSHIBA J3100SS Dyna Book
定価¥198,000
➡特価¥149,000

各種特選・特価品

高機能関数ポケットコンピュータ
PC-E550　〈新製品〉
¥32,000➡特価！

64KバイトRAMを標準装備
●関数電卓モード搭載
●充実の124関数機能

PC-500と各種パソコンをつなぐインターフェースケーブル
CE-140T　¥8,800

パソコンファクス MZ-1V01　〈限定セット販売！〉
●MZ25セット（インターフェースソフト付）
標準価格合計¥342,800➡¥120,000
●MZ-1V01（本体のみ）
標準価格合計¥278,000➡¥98,000

漢字カラー熱転写プリンタ
シャープMZ-1P22
標準価格¥59,800➡特価¥25,000

キヤノンLASER SHOTプリンタ
LBP-B406S
定価¥498,000➡特価！
LBP-A404S
定価¥265,000➡特価！

シャープMZ-1X30 モデムホン
（1×19上位機種）　〈在庫限り〉
標準価格¥98,000➡¥29,800

ビデオボード CZ-6BV1
¥21,000➡特価

FM TOWNS お買い得セット

1. TOWNS-1（本体）	¥338,000
2. FMT-ME 1M（増設メモリー）	¥60,000
3. FMD-FD301（増設FDユニット）	¥28,000
4. FMT-KB101（キーボード）	¥20,000
5. FMT-DP531（カラーディスプレイ）	¥89,800
6. TOWNS-OS V1.1 L20	¥20,000
定価合計	¥555,800

大特価／¥285,000

〈全商品新品完全保証付〉■シャープポケコン全商品販売中。カタログ、特価表ご請求ください（〒72）

☎0426-45-3001～3
FAX.0426-44-6002
●営業時間/10:00～19:00●電話受付/20:00迄可●定休日/日曜日（祭日営業）
SHARP SUPER XEX SHOP
アイビット電子株式会社　〒192 東京都八王子市北野町560-5

上記の広告商品はすべて店頭販売もしております。

全国通信販売
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店　（普）1752505

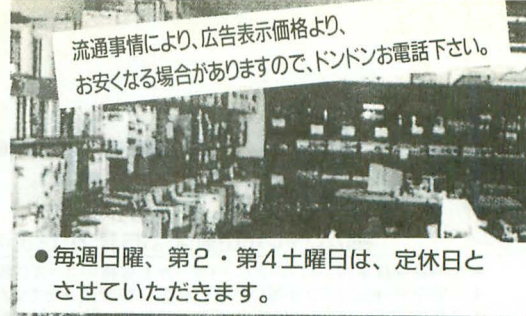
●本誌発売時には上記価格よりさらにお求めやすい価格に変更されている場合があります。●この広告の商品にはすべて送料・消費税は含まれておりません。

オーエーブレイン 全国通販

●オフコンからパソコンまで
幅広～い品揃え。おまかせあれ!! お電話くださいネ! ☎03-5688-3621

★全商品保証書付。専門のアドバイザーがお客様のニーズに親切に対応します。
★初期不良・輸送トラブル等に迅速に対応し、即交換させていただきます。
★送料は1個につき¥1,000です。(※一部離島は除きます。お問合せ下さい。)

●ご注文、お問合せは…毎日午前10時から午後8時まで
●下取・買取は電話で見積りしております。責任を持って下取りさせて頂きま
●商品のお届けは…入金確認後、即日発送致します。

OAB特選～X68000シリーズセット (ゲームパック・ディスケット付) (税抜き)

①X68000 EXPERTII
●CZ-603C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計¥453,000

●SX-WINDOW搭載!!

クレジット例

1回	¥345,000	12回	¥30,200×12

OAB大特価

②X68000 EXPERTII-HD
●CZ-613C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計¥563,000

クレジット例

1回	¥428,000	12回	¥37,500×12

OAB大特価

③X68000 PROII
●SX-WINDOW搭載!!
●CZ-653C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計¥400,000

クレジット例

1回	¥297,000
12回	¥26,000×12

OAB大特価

④X68000 PROII-HD
●CZ-663C-BK/GY
●CZ-605D-BK/GY
●MD-2HD 20枚
定価合計¥510,000

OAB大特価

⑤X68000 SUPER-HD
X68000 SUPER-HD ●SX-WINDOW搭載!!
●SX-WINDOW搭載
●SCSIインターフェース装備
●80MBハードディスク搭載
●3MB大容量メモリ装備
●高解像度グラフィック

●CZ-623C-TN(チタン)
●CZ-613D-TN(チタン)
●MD-2HD 20枚
定価合計¥633,000

クレジット例

1回	¥485,000	12回	¥42,000×12

OAB大特価

X68000 EXPERT-HD=特選限定品

X68000 EXPERT-HD

●CZ-612(BK)
(定価¥466,000)
●CZ-605(BK)
(定価¥115,000)
定価合計¥581,000

OAB特価¥368,000

OAB特選セット(中古美品セット)
①CZ-623C-TN+CZ-602D
2台限り…………¥460,000
②CZ-613C+CZ-605D
1台限り…………¥358,000
③CZ-612C+CZ-600D
3台限り…………¥348,000

周辺機器コーナー

プリンターセットコーナー
●CZ-6PVI(カラービデオプリンター)
定価¥198,000…………▶特価¥152,000
●CZ-8PC3(24ドット熱転写カラープリンター)
定価¥65,800…………▶特価¥53,000
●CZ-8PK10(24ピン漢字ドットプリンター・136桁)
定価¥97,800…………▶特価¥73,000
●CZ-8PGI(24ピンカラー漢字ドットプリンター・80桁)
定価¥130,000…………▶特価¥98,000
●CZ-8PG2(24ピンカラー漢字ドットプリンター・136桁)
定価¥160,000…………▶特価¥119,000
●IO-735X(カラーイメージェットプリンター)
定価¥248,000…………▶特価¥185,000

■CZ-8PC4(定価¥99,800)
特選品!!
●48ドット熱転写カラー
漢字プリンター

特価¥64,800

X68000用ソフトウェアー・コーナー
①CZ-212BS(BUSINESS)…………定価¥68,000▶特価¥53,000
②CZ-220BS(DATA)…………定価¥58,000▶特価¥45,000
③CZ-215MS(Sampling)…………定価¥17,800▶特価¥13,800
④CZ-221HS(NEW Print Shop)……定価¥10,800▶特価¥15,500
⑤CZ-227BS(TOP財務会計)………定価¥200,000▶特価¥158,000
⑥CZ-226BS(CARD)…………定価¥229,800▶特価¥23,000
⑦CZ-223CS(Communicabion)………定価¥19,800▶特価¥115,500
⑧CZ-213MS(MUSIC)…………定価¥18,800▶特価¥14,800
⑨CZ-211LS(C compiler)…………定価¥39,800▶特価¥31,000
⑩C-TRACE(キャスト)…………定価¥68,000▶特価¥52,000
⑪EW(イースト)…………定価¥38,000▶特価¥29,000

X68000用周辺機器コーナー
●CZ-6BEIB… 定価¥28,000▶特価¥22,000
●CZ-6BMI…… 定価¥26,800▶特価¥21,000
●CZ-6EBI…… 定価¥88,000▶特価¥69,800
●CZ-6VTI…… 定価¥69,800▶TEL下さい
●CZ-8NSI…… 定価¥188,000▶特価¥149,000
●CZ-6BCI…… 定供¥79,800▶特価¥63,000

今月の特価品(限定)お早目に!!
★CZ-652C(BK)+CZ-602D(BK)
4セット限り …… 大特価¥258,000
●SHARP WD-A300(ワープロ)
定価¥165,000…………特価¥110,000
●SHARP WD-A330(ワープロ)
定価¥185,000…………特価¥125,000
●SHARP WD+HL30(ワープロ)
定価¥198,000…………特価¥134,000

●SHARP PW-910(ワープロ)
…………特価¥85,000
●NEC PC-KD853(アナログCRT)
…………特価¥50,000
●三菱XC-1498C(アナログCRT)
…………特価¥54,800
●SHARP CU-14FD(アナログCRT)
…………特価¥46,000
●SHARP PA-8500(電子手帳)
…………特価¥16,000

I・O DATA 増設RAMボード
●1MB増設PAMボード
PIO-6BEI-A
定価
¥25,000
特価¥18,800

●2MB増設RAMボード
PIO-6BE2-2M
定価
¥50,000
特価¥37,800

●4MB増設RAMボード
PIO-6BE4-4M
定価
¥88,000
特価¥65,800

■ハードディスク ■特価品もありますのでTEL下さい。
●アイテック ITX-640…………特価¥117,000
●アイテック ITX-680…………特価¥149,000
●ロジテック LHD-32V…………特価¥85,000
●ロジテック LHD-34VE…………特価¥90,000
●ロジテック LHD-34V…………特価¥104,000
●シャープ CZ-620H…………特価¥118,000
●シャープ CZ-64H…………特価¥95,000
●アイテム HXD-040…………特価¥88,000
●アイテム HXD-042…………特価¥95,000
●ICM SR 80…………特価¥130,000

中古パソコン (価格/在庫は変動します。予約は5日以内とします。)

機種	価格	機種	価格
PC-9801RA5	¥338,000より	PC-286VS	¥165,000より
PC-9801RA2	¥265,000より	CZ-600C	¥160,000より
PC-9801RX2	¥199,000より	CZ-601C	¥170,000より
PC-9801EX2	¥190,000より	CZ-611C	¥198,000より
PC-9801VX21	¥170,000より	CZ-652C	¥178,000より
PC-9801UX21	¥165,000より	CZ-612C	¥210,000より
PC-9801VX2	¥160,000より	68000用モニター	¥49,000より
PC-9801VM21	¥150,000より	PC-9801用サウンドボード	¥13,000より
PC-9801UV11	¥148,000より	PC-88SR, FR	¥50,000より
PC-9801LV22	¥160,000より	PC-88FH, FA	¥65,000より
PC-286VE	¥150,000より	400ラインCRT	¥38,000より
PC-286US	¥155,000より	200ラインCRT	¥10,000より

オーエーブレイン今月の特価品!! 台数限定 お早目に!!

ドライブ・ユニット
アクセル
●FDC-357……特価¥36,000
●FDC-358……特価¥49,000
コンピュータ・リサーチ
●CRC-FD3.5S…特価¥29,000
●CRC-FD3.5W…特価¥42,000
グローリア
●GD-35MI……特価¥23,000
●GD-35M2……特価¥39,000
緑電子
●Little-FI……特価¥26,000
●Little-F2……特価¥38,000
SNE
●SNE-2……特価¥49,000

プリンター
NEC
●PC-PR201G+
PC-PR201G-04……特価¥99,800
●NM-4150……特価¥132,000
エプソン
●AP-850PC……特価¥64,000
●VP-2050PC……特価¥99,000
キャノン
●BJ-130J……特価¥125,000

ハード・デイスク

■ARK WOOD
NEC純正ドライブ
●AW-N40C
(定価¥1
…特価¥9
●AW-N100
(定価¥1
…特価¥13

サウンド・ボード
SNE
1 サウンドオーケストラV……特価¥23,000
2 サウンドオーケストラL……特価¥17,800
3 リトルオーケストラ……特価¥14,500
4 リトルオーケストラL……特価¥12,000
5 サウンド・ミュージシャン…特価¥16,000
6 リトルミュージシャン……特価¥8,500

ワープロ
NEC
●PWP-70HR……特価¥17
●PWP-70R……特価¥12
●PWP-50RD……特価¥11
東芝
特価¥14
●JW-95HD……特価¥14
●JW-95F……特価¥10
キャノン
●CW-α350……特価¥15
その他、シャープ、松下、富士通等、T

通信販売によるご購入方法(お電話でお申し込み下さい。)

現金一括払い	クレジット	振込先
銀行振込：電信扱いにてお振込下さい 手数料はお客様負担となります 現金書留：住所、氏名、電話番号、商品名、使用機種、メディア等をお書き添えのうえ、現金書留にて当社までお送り下さい	専用のお申し込み用紙をお送り致しますので、必要事項をご記入・捺印のうえ、ご返送下さい ※未成年者の方は、保護者のご承認を受けてからお申し込み下さい	●第一勧業銀行 御徒町支店 (普)1376679 オーエーブレイン ●朝日信用金庫 本店 (普)334833 オーエーブレイン

★クレジットは1～60回払いで月々5,000円よりご自由に設定できます。

オーエーブレイン 〒110 東京都台東区台東1-28-4
TEL & FAX 5688-3621

■流通事情により、広告表示よりお安くなる場合もございます。まずは、お電話下さい。■ビジネス・ゲームセットもございます。

ACCEL TOMATO series ICM LAND OMRON 株式会社メルコ maxell I·O DATA SONY FUJI FILM Caravelle WINTECH

Sofmap 商品は今すぐ！お支払いは9ヶ月後から。

For Computer Communication Age

株式会社ソフマップ

冬のボーナス一括払い金利0(ゼロ)受け付け開始!!(10/1〜) ボーナスのみ1·2·4·6·8·10回払いもOK!

●この表の価格は8月30日現在のものです。●掲載価格には消費税が含まれておりません。

X68000 SUPER-HD EXPERTII・II-HD

SUPERセット SUPER-HD

大容量80MB、3.5インチHD内蔵、SCSIインターフェイス標準装備、SX-WINDOW搭載

月々¥5,900より

クレジット注文NO.1

商品	価格
CZ-623C-TN〈本体〉	¥Sofmap特価
CZ-613D-TN〈15″ドットピッチ0.31〉	¥Sofmap特価
CZ-8PC4〈48ドット熱転写プリンタ〉	¥Sofmap特価
CZ-8NJ2〈アナログスティック〉	¥Sofmap特価
CZ-6BF1〈増設用RS-232Cボード〉	¥Sofmap特価
CZ-6BM1〈MIDIボード〉	¥Sofmap特価
CM-64〈音源モジュール〉	¥Sofmap特価
AN-S100〈アンプ内蔵スピーカー〉	¥Sofmap特価
MD-24FS5〈通信用モデム2400BPS〉	¥Sofmap特価
GT-6000〈フルカラーイメージスキャナー〉	¥Sofmap特価
♯5220〈RS-232Cケーブル〉	¥Sofmap特価
CZ-257CS〈Communication PRO-68K Ver2.0〉	¥Sofmap特価
CZ-252MS〈Music studio PRO-68K Ver1.1〉	¥Sofmap特価
CZ-211LS〈C compiler PRO-68K〉	¥Sofmap特価
CZ-219SS〈OS-9〉	¥Sofmap特価
CZ-251BS〈Hyperword〉	¥Sofmap特価
Z's STAFF〈PRO-68K Ver2.0〉	¥Sofmap特価
ゲームソフト2本〈定価¥9,800以下のお好きなソフト〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥1,470,000	¥お電話にて

月々	ボーナス
¥ 5,900×84回	ボーナス¥ 80,000×14回
¥ 8,000×72回	ボーナス¥ 80,000×12回
¥10,000×60回	ボーナス¥ 85,000×10回
¥12,100×48回	ボーナス¥100,000× 8回
¥15,000×36回	ボーナス¥130,000× 6回

基本セット SUPER-HD

集積度を高めた"マンハッタンシェイプ"3Mの大容量メモリSX-WINDOW搭載

月々¥2,400から

クレジット注文NO.2

商品	価格
CZ-623C-TN〈本体〉	¥Sofmap特価
CZ-613D-TN〈15″ドットピッチ0.31〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥634,000	¥お電話にて

月々	ボーナス
¥ 2,400×72回	ボーナス ¥40,000×12回
¥ 4,400×60回	ボーナス ¥35,000×10回
¥ 8,200×84回	ボーナス なし
¥11,300×54回	ボーナス なし
¥15,600×36回	ボーナス なし

開発セット SUPER-HD

3Mバイトの大容量メモリ、拡張I/Oボード4スロット標準装備

月々¥2,100から

クレジット注文NO.3

商品	価格
CZ-623C-TN〈本体〉	¥Sofmap特価
CZ-613D-TN〈15″ドットピッチ0.31〉	¥Sofmap特価
CZ-8PK10〈24ピンプリンター130桁〉	¥Sofmap特価
CZ-211LS〈C compiler PRO-68K〉	¥Sofmap特価
CZ-219SS〈OS-9〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥801,400	¥お電話にて

月々	ボーナス
¥ 2,100×84回	ボーナス ¥40,000×14回
¥ 4,700×48回	ボーナス ¥50,000× 8回
¥ 9,700×72回	ボーナス なし
¥13,000×48回	ボーナス なし
¥16,600×36回	ボーナス なし

基本セット EXPERTII-HD

月々¥1,700から

クレジット注文NO.4

商品	価格
CZ-613C-BK〈本体〉	¥Sofmap特価
CZ-605D-BK〈15″ドットピッチ0.39〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥564,000	¥お電話にて

月々	ボーナス
¥ 1,700×72回	ボーナス ¥40,000×12回
¥ 4,400×48回	ボーナス ¥40,000× 8回
¥ 7,500×84回	ボーナス なし
¥11,100×48回	ボーナス なし
¥14,200×36回	ボーナス なし

グラフィックセット EXPERTII-HD

月々¥2,100から

クレジット注文NO.5

商品	価格
CZ-613C-BK〈本体〉	¥Sofmap特価
CZ-613D-BK〈15″ドットピッチ0.31〉	¥Sofmap特価
CZ-8PC4〈48ドット熱転写プリンタ〉	¥Sofmap特価
GT-6000〈フルカラーイメージスキャナー〉	¥Sofmap特価
♯5220〈RS-232Cケーブル〉	¥Sofmap特価
Z's STAFF PRO-68K Ver2.0	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥927,300	¥お電話にて

月々	ボーナス
¥ 2,100×84回	ボーナス ¥60,000×14回
¥ 5,200×60回	ボーナス ¥60,000×10回
¥ 9,800×36回	ボーナス ¥80,000× 6回
¥12,200×84回	ボーナス なし
¥15,200×60回	ボーナス なし

基本セット EXPERTII

月々¥2,200から

クレジット注文NO.6

商品	価格
CZ-603C〈本体〉	¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥454,000	¥お電話にて

月々	ボーナス
¥ 2,200×48回	ボーナス ¥40,000× 8回
¥ 5,900×84回	ボーナス なし
¥ 8,200×54回	ボーナス なし
¥11,300×36回	ボーナス なし
¥16,200×24回	ボーナス なし

ゲームセット EXPERTII

月々¥2,500から

クレジット注文NO.7

商品	価格
CZ-603C〈本体〉	¥Sofmap特価
CZ-613D〈15″ドットピッチ0.31〉	¥Sofmap特価
CZ-8NJ2〈アナログスティック〉	¥Sofmap特価
ゲームソフト2本〈定価¥9,800以下のお好きなソフト〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥517,400	¥お電話にて

月々	ボーナス
¥ 2,500×48回	ボーナス ¥45,000× 8回
¥ 4,400×36回	ボーナス ¥50,000× 6回
¥ 7,500×72回	ボーナス なし
¥10,000×48回	ボーナス なし
¥12,800×36回	ボーナス なし

ワープロセット EXPERTII

月々¥2,600から

クレジット注文NO.8

商品	価格
CZ-603C〈本体〉	¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉	¥Sofmap特価
CZ-8PK10〈24ピンプリンター130桁〉	¥Sofmap特価
CZ-251BS〈Hyperword〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥591,600	¥お電話にて

月々	ボーナス
¥ 2,600×84回	ボーナス ¥30,000×14回
¥ 7,700×84回	ボーナス なし
¥ 9,700×60回	ボーナス なし
¥11,500×48回	ボーナス なし
¥14,700×36回	ボーナス なし

MIDIセット EXPERTII

月々¥2,500から

クレジット注文NO.9

商品	価格
CZ-603C〈本体〉	¥Sofmap特価
CZ-613D〈15″ドットピッチ0.31〉	¥Sofmap特価
CZ-6BM1〈MIDIボード〉	¥Sofmap特価
CM-64〈音源モジュール〉	¥Sofmap特価
AN-S100〈アンプ内蔵スピーカー〉	¥Sofmap特価
CZ-247MS〈MUSIC PRO-68K(MIDI)〉	¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉	¥Sofmap特価
標準価格¥695,200	¥お電話にて

月々	ボーナス
¥ 2,500×84回	ボーナス ¥40,000×14回
¥ 4,000×60回	ボーナス ¥45,000×10回
¥ 9,200×84回	ボーナス なし
¥13,700×48回	ボーナス なし
¥17,600×36回	ボーナス なし

プリンター

CZ-8PC4
定価¥99,800
¥お電話にて

CZ-8PK10
定価¥97,800
¥お電話にて

CZ-8PG1
定価¥130,000
¥お電話にて

CZ-8PG2
定価¥160,000
¥お電話にて

周辺機器

商品	定価	ソフマップ特価
●PIO-6BE1-A	¥ 25,000	¥18,200
●PIO-6BE2-2M	¥ 50,000	¥36,800
●PIO-6BE4-4M	¥ 88,000	¥64,800
●CZ-6BE4	¥138,000	¥お電話にて
●CZ-6BF1	¥ 49,800	¥お電話にて
●CZ-6BP1	¥ 79,800	¥お電話にて
●CZ-6TU	¥ 33,100	¥お電話にて
●AN-S100	¥ 36,600	¥お電話にて
●CZ-8NS1	¥188,000	¥お電話にて
●CZ-6EB1	¥ 88,000	¥お電話にて

SOFT WARE

商品	定価	ソフマップ特価
●Zs STAFF PRO68K V2.0	¥58,000	¥お電話にて
●DATA PRO68K(CZ-220BS)	¥58,000	¥お電話にて
●CARD PRO68K(CZ-226BS)	¥29,800	¥お電話にて
●Cコンパイラ PRO68K V2.0(CZ-245LS)	¥39,800	¥お電話にて
●SOUND PRO68K(CZ-214MS)	¥15,800	¥お電話にて
●MUSIC PRO68K(CZ-213MS)	¥15,800	¥お電話にて
●サンプリング PRO68K(CZ-215MS)	¥17,800	¥お電話にて
●コミュニケイション V2.0(CZ-257CS)	¥19,800	¥お電話にて
●OS-9(CZ-219SS)	¥29,800	¥お電話にて

●各種ゲームソフト ····定価より15〜20%OFF

No.1システム

No.1 下取りシステム

お持ちの機種を下取りに出して、新品に買替えようと思っている方、ソフマップに御相談下さい。
買取り価格がどこよりも高く、新品の販売価格がどこよりも安いから、当然どこよりもお得な条件でお買求めいただけます。
又、差額を商品券でお支払いもできます。

No.1 配送システム

1.到着日指定、夜間配送システム
お客様のご都合に合わせて配送させていただきます。機種によっては、夜間配送できないものがあります。

2.代金引換システム(要手数料)
係員が品物をお届けに行きますので、その時にお支払い下さい。

No.1 クレジットシステム

1. 9ヶ月先からのお支払いOK
スキップクレジットを御利用になれば支払い開始月を1ヶ月から、最長9ヶ月先までおくらせる事が出来ます。

2. 月々¥1,000からのお支払いOK
月々のお支払い金額の設定が¥1,000からOK。

3. 84回払いもOK
お客様のプランに合わせて、1回から最長84回まで支払い回数をお選びいただけます。

4. ステップアップクレジット
お客様のプランに合わせて、毎月のお支払い金額を徐々に増やしていくシステムです。例えば、1年目は¥3,000、2年目は¥6,000というように、御自由に設定することができます。

5. ボーナス10回払いもOK
毎月の支払いは0(ゼロ)、ボーナス時のみのお支払いでクレジットが御利用になれます。回数は1、2回の他、4·6·8·10回払いまでOK。

6. カードクレジット
各種クレジットカードが店頭だけでなく、通信販売でも御利用になれます。詳しくはお気軽にお問い合わせ下さい。

7. カレッジクレジット
保証人なしで、学生の方でもクレジットが御利用できます。(20歳以上)

No.1 サポートシステム

1. 初期不良交換期間3ヶ月
●万一、お届けした商品が不良の場合、お買い上げ日より3ヶ月以内なら、同等品と即、交換致します。

2. 新品パソコン3年保証
●メーカー保証が1年の場合、メーカー保証1年＋マップ保証2年の計3年間の保証になります。

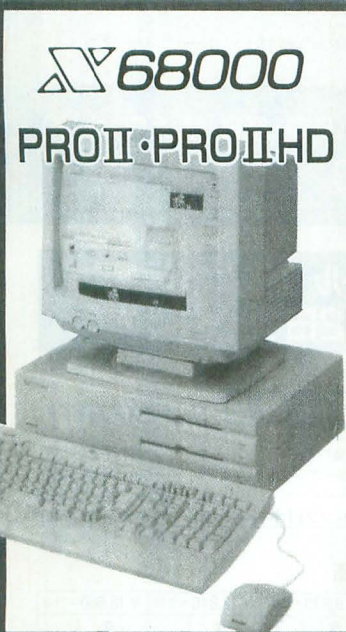
3. 中古パソコン1年保証
●中古パソコン本体は、1年間保証致します。(ディスプレイプリンタ等は6ヶ月保証となります)

4. 新品パソコン買取り保証
●1ヶ月以内であれば必ず買取り保証金額で、下取り、買取り致します。

5. 永久買取り保証
●古くなったパソコン、スクラップ寸前のパソコンでもOK!! どんなパソコンでも、どこよりも高く買い取ります。

ACCEL TOMATO series ICM LAND OMRON 株式会社メルコ maxell I-O DATA SONY FUJI FILM Caravelle WINTECH

下取り・買取り大歓迎!!

TV 日本テレビ、TBS、フジテレビ、テレビ朝日、テレビ東京系列でCM放映中!!直営10店舗

下取り差額のお支払いは、クレジットを御利用下さい。

基本セット PROII-HD　月々¥2,300から

クレジット注文NO.10

CZ-663C〈本体〉……¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥511,000　¥お電話にて

支払	ボーナス
¥2,300×72回	ボーナス ¥30,000×12回
¥4,900×60回	ボーナス ¥20,000×10回
¥8,300×60回	ボーナス なし
¥12,600×36回	ボーナス なし
¥18,100×24回	ボーナス なし

ビジネスセット PROII-HD　月々¥2,300から

クレジット注文NO.11

CZ-663C〈本体〉……¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
CZ-8PG2〈24ピン漢字ドットプリンター130桁〉……¥Sofmap特価
CZ-212BS〈BUSINESS PRO-68K〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥739,000　¥お電話にて

支払	ボーナス
¥2,300×72回	ボーナス ¥50,000×12回
¥6,100×48回	ボーナス ¥50,000× 8 回
¥9,700×84回	ボーナス なし
¥12,000×60回	ボーナス なし
¥14,400×48回	ボーナス なし

データベースセット PROII-HD　月々¥2,300から

クレジット注文NO.12

CZ-663C〈本体〉……¥Sofmap特価
CZ-613D〈15″ドットピッチ0.31〉……¥Sofmap特価
CZ-8PG1〈24ピン漢字ドットプリンター80桁〉……¥Sofmap特価
CZ-226BS〈CARD PRO-68K〉……¥Sofmap特価
CZ-220BS〈DATA PRO-68K〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥748,800　¥お電話にて

支払	ボーナス
¥2,300×84回	ボーナス ¥45,000×14回
¥5,500×60回	ボーナス ¥40,000×10回
¥9,800×84回	ボーナス なし
¥13,500×54回	ボーナス なし
¥18,600×36回	ボーナス なし

基本セット PROII　月々¥3,200から

クレジット注文NO.13

CZ-653C〈本体〉……¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥401,000　¥お電話にて

支払	ボーナス
¥3,200×60回	ボーナス ¥20,000×10回
¥5,300×84回	ボーナス なし
¥7,800×48回	ボーナス なし
¥10,000×36回	ボーナス なし
¥14,400×24回	ボーナス なし

通信セット PROII　月々¥1,800から

クレジット注文NO.14

CZ-653C〈本体〉……¥Sofmap特価
CZ-613D〈15″ドットピッチ0.31〉……¥Sofmap特価
CZ-8PG1〈24ピン漢字ドットプリンター80桁〉……¥Sofmap特価
MD-24FS5〈通信モデム2400BPS〉……¥Sofmap特価
CZ-257CS〈Communication PRO-68K Ver2.0〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥620,600　¥お電話にて

支払	ボーナス
¥1,800×60回	ボーナス ¥50,000×10回
¥5,400×36回	ボーナス ¥60,000× 6 回
¥9,000×72回	ボーナス なし
¥12,100×48回	ボーナス なし
¥15,400×36回	ボーナス なし

プリントセット PROII　月々¥2,200から

クレジット注文NO.15

CZ-653C〈本体〉……¥Sofmap特価
CZ-605D〈15″ドットピッチ0.39〉……¥Sofmap特価
CZ-8PC4〈48ドット熱転写プリンタ〉……¥Sofmap特価
CZ-221HS〈NEW Printshop PRO-68K〉……¥Sofmap特価
CZ-235GS〈グラフィックライブラリVOL.1〉……¥Sofmap特価
CZ-236GS〈グラフィックライブラリVOL.2〉……¥Sofmap特価
マクセルブランクディスケット〈5″2HD×10枚〉¥Sofmap特価
標準価格¥538,200　¥お電話にて

支払	ボーナス
¥2,200×60回	ボーナス ¥40,000×10回
¥4,700×48回	ボーナス ¥35,000× 8 回
¥7,100×84回	ボーナス なし
¥9,800×54回	ボーナス なし
¥13,500×36回	ボーナス なし

●掲載の商品以外にも多数取り扱いしておりますので、お気軽にお問い合わせ下さい。又、商品在庫は毎日変動しますので、品切れの際は御予約承ります。

よりもお得しかない!額クレジットOK!!

高額下取り差額表

下取り差額は随時変動します。この他の商品についてもお電話でお気軽にお問い合わせ下さい。

お送りになる方、又は直接東京店に来られる方 0120-110-994
直接大阪店に来られる方 ☎06-641-8801
商品の先り先 〒101 千代田区外神田1-8-3 野水ビル3F ソフマップ買取りセンター

あなたが今、欲しい機種（新品）

あなたが今、お持ちの機種 下取り機種	SUPER-HD CZ-623C CZ-613D 交換差額	EXPERT II CZ-603C CZ-605D 交換差額	EXPERTIIHD CZ-613C CZ-613D 交換差額	PROII CZ-653C CZ-605D 交換差額	PROII-HD CZ-603C CZ-605D 交換差額
CZ-652C CZ-602D	¥308,000	¥135,000	¥238,000	¥ 95,000	¥135,000
CZ-602C CZ-602D	¥268,000	¥ 95,000	¥198,000	¥ 55,000	¥ 95,000
CZ-611C CZ-611D	¥270,000	¥ 97,000	¥200,000	¥ 57,000	¥ 97,000
CZ-601C CZ-601D	¥318,000	¥145,000	¥248,000	¥105,000	¥145,000
CZ-600C CZ-601D	¥323,000	¥150,000	¥253,000	¥110,000	¥150,000
CZ-880C CZ-880D	¥440,000	¥267,000	¥370,000	¥227,000	¥267,000
PC-9801VX21 PC-KD854N	¥313,000	¥140,000	¥243,000	¥100,000	¥140,000
FM-TOWNS-2 FMT-DP531	¥393,000	¥220,000	¥323,000	¥180,000	¥220,000

高額買取価格表

商品名	高額買取
X68000 モニターセット	
X68(CZ-662C+CZ-600D/601D)	¥250,000
X68(CZ-662C+CZ-611D/612D)	¥260,000
X68(CZ-652C+CZ-600D/601D)	¥210,000
X68(CZ-652C+CZ-611D/612D)	¥220,000
X68(CZ-623C+CZ-602D)	¥360,000
X68(CZ-623C+CZ-605D)	¥380,000
X68(CZ-623C+CZ-613D)	¥390,000
X68(CZ-623C+CZ-603D)	¥345,000
X68(CZ-623C+CZ-604D)	¥350,000
X68(CZ-612C+CZ-600D/601D)	¥290,000
X68(CZ-612C+CZ-611D/612D)	¥300,000
X68(CZ-611C+CZ-600D/601D)	¥235,000
X68(CZ-611C+CZ-611D/612D)	¥245,000
X68(CZ-603C+CZ-602D)	¥255,000
X68(CZ-603C+CZ-605D)	¥270,000
X68(CZ-603C+CZ-613D)	¥280,000
X68(CZ-603C+CZ-603D)	¥215,000
X68(CZ-603C+CZ-604D)	¥225,000
X68(CZ-602C+CZ-600D/601D)	¥240,000
X68(CZ-602C+CZ-611D/612D)	¥250,000
X68(CZ-601C+CZ-600D/601D)	¥195,000
X68(CZ-600C+CZ-600D/601D)	¥190,000

業界No.1の低金利

支払回数	1	3	6	10	12	15	18	20	24	30	36	42	48	54	60	66	72	78	84
他社金利	3	4	5	7	9	10	12	13	16	19	21	25	28	31	35	−	−	−	−
Sofmap金利	2.0	3.0	4.0	5.5	5.5	8.5	11.0	11.0	11.5	16.0	16.0	20	21	26	27	33	35	39	42

お支払い方法

1. 代金引換システム
係員が品物をお届けに行きますので、その時にお支払い下さい
2. クレジット
お電話で支払い回数、支払い開始日、ボーナスの有無をおっしゃって下さい
こちらからクレジット用紙をお送り致しますので、ご記入・ご捺印の上ご返送下さい
商品到着後、御指定の口座から自動引落しとなります
3. 銀行振込
お電話で御注文の上、下記振込先へ電信扱いでお振り込み下さい
ご確認後、ただちに商品をお送りします 振込手数料はお客様負担となります

東京秋葉原店

大阪日本橋店 地下鉄恵美須町駅より徒歩1分

店頭に直接来られる方は	☎東京03-258-3156 ☎大阪06-647-0562
通信販売をご利用の方は	☎東京03-253-4230 ☎大阪06-633-7224 FAX.03-253-4290 札幌011-865-7030 仙台022-268-3405 新潟0252-22-6139 横浜045-311-3441 金沢0762-21-7045 名古屋052-332-2117 広島082-222-0604 福岡092-752-0044 高松0878-34-8833
24時間テレフォンサービス	☎03-258-7910
フリーダイヤル 商品発送のお問合わせ	0120-08-0113
フリーダイヤル 故障・修理のお問合わせ	0120-11-0292

振込先
東京秋葉原店
三和銀行秋葉原支店(普)1012131
口座名義 (株)ソフマップ

営業時間／年中無休 ●平日AM11:00〜PM8:00 ●日・祭日AM10:00〜PM7:00

〒101 東京都千代田区外神田3丁目15番6号小暮末広ビル1F
〒556 大阪市浪速区日本橋5丁目7番17号ソフマップビル

クリエイト特典

- ●全商品完全保証書付(メーカー保証)
- ●全国無料配達(一部離島の方は有料になります)
- ●配達日の指定OK(日曜・祭日にかかわらずお客様のご都合にあわせて配達します)
- ●どんな商品の組合せも自由自在(ご予算、用途に応じ自由自在にシステムアップできます)
- ●中古パソコン高額下取り(今お使いのパソコンをわずかな差額でグレードアップ)
- ●お支払い方法自由(低金利の均等払い、ボーナス一括払いもご利用ください)

営業時間(定休日▶渋谷店:日曜・祭日/横浜店:水曜)
AM10:00~PM7:00

当社はX68000の販売認定店です。どんなことでも安心してご相談ください。

『渋谷店大改装謝恩セール』開催!
9月25日(火)➡10月2日(火)

X68000 NEW PRO II

- ●CZ-653C(本体)……………………¥285,000
- ●CZ-603D(カラーディスプレイ)……………………¥ 84,800
- ●お好きなゲームソフト1本……………………¥ 7,800
- ■定価合計……………………¥377,600

クリエイト特価

均等払い	¥ 7,680×48回	¥ 9,890×36回	¥14,370×24回
ボーナス	なし	なし	なし

X68000 EXPERTセット

5 台 限 定 !

- ●CZ-602C-GY(本体)……………………¥356,000
- ●CZ-603D-GY(カラーディスプレイ)……………………¥ 84,800
- ■定価合計……………………¥440,800▶大特価¥279,000

大特価¥279,000

均等払い	¥12,850×24回	¥ 8,870×36回	¥ 6,920×48回
ボーナス	なし	なし	なし

X68000 NEW EXPERT II

- ●CZ-603C(本体)……………………¥338,000
- ●CZ-613D(カラーディスプレイテレビ)……………………¥135,000
- ●CZ-8NJ2……………………¥ 23,800
- ●お好きなゲームソフト1本……………………¥ 9,800
- ■定価合計……………………¥506,600

クリエイト特価

均等払い	¥ 9,970×48回	¥12,840×36回	¥18,660×24回
ボーナス	なし	なし	なし

X68000 SUPER HD

- ●CZ-623C-TN(本体・キーボード・マウス)……………………¥498,000
- ●CZ-613D-TN(カラーディスプレイ)……………………¥135,000
- ●CZ-6BP1……………………¥ 79,800
- ■定価合計……………………¥712,800

クリエイト特価

均等払い	¥ 7,320×48回	¥10,100×36回	¥13,450×24回
ボーナス	¥42,000×8回	¥50,000×6回	¥80,000×4回

※本広告に掲載の全商品の価格について消費税は含まれておりません。

X68000 NEW EXPERT II

ミュージシャンセット。これもTMネットワークだよ~!

- ●CZ-603C……………………¥338,000
- ●CZ-605D……………………¥115,000
- ●MU1.B(MIDIボード&ソフト)……………………¥ 39,800
- ●CM32L……………………¥ 69,000
- ●グラナダ……………………¥ 8,800
- ●JOYカード……………………¥ 1,800
- ■定価合計……………………¥572,400▶超特価¥458,000

X68000 NEW PRO II

ゲーマーズセット。遊んで暮らせるSET!

- ●PRO II CZ653C……………………¥285,000
- ●0.31CRT CZ603D……………………¥ 84,800
- ●グラナダ……………………¥ 8,800
- ●Y'S……………………¥ 8,700
- ●ポピュラス……………………¥ 9,800
- ●スーパーハングオン……………………¥ 8,800
- ●エージャックス……………………¥ 8,800
- ●サーク……………………¥ 8,800
- ●アールタイプ……………………¥ 7,800
- ●アナログJOYSTIC XE-1AP……………………¥ 13,800
- ■定価合計……………………¥445,100▶超特価¥353,000

★この表以外の組合せ、お支払い方法もご自由にできます。
★X1シリーズ用、X68000シリーズ用各社ハードディスク/プリンタ等の周辺機器を大特価にて販売しております。
電話にてお問合せください。

X68000シリーズ用 周辺機器・ソフト オール超特価!!

型番	品名	定価	ソフト名	品名	定価
CZ-6VT1	カラーイメージユニット	¥ 69,800	MUSIC PRO	MIDI版	¥ 28,800
CZ-8NS1	カラーイメージスキャナ	¥188,000	MUSIC PRO-68K	マウスを使った楽譜ワープロ	¥ 18,800
CZ-6BE1A	IMB増設RAMボード	¥ 38,000	SOUND PRO-68K	サウンドエディタ	¥ 15,800
CZ-6BE2	2MB増設RAMボード	¥ 79,800	Sampling PRO-68K	AD PCMサンプリングエディタ	¥ 17,800
CZ-6BE4	4MB増設RAMボード	¥138,000	Musicstudio PRO-68K V.1.1	MIDIマルチレコーディングソフト	¥ 28,800
CZ-8NM3	マウス・トラックボール	¥ 9,800	OS-9/X68000	マルチタスクオペレーティングシステム	¥ 29,800
BF-68PRO	高性能CRTフィルター	¥ 19,800	PRO-68K	サイバーノート	¥ 19,800
CZ-6BP1	数値演算プロセッサ・ボード	¥ 79,800	PRO-68K	ステーショナリー	¥ 14,800
CZ-8NT1	トラックボール	¥ 13,800	Ccompiler PRO-68K	ソフト開発セット	¥ 39,800
CZ-6BM1	MIDIボード	¥ 26,800	Human 68K Ver2.0	開発ツールセット	¥ 9,800
CZ-8NJ2	アナログスティック	¥ 23,800	PIO-6BE1-A	内蔵1MRAM	¥ 25,000
CZ-6TU	パソコンチューナ	¥ 33,100	PIO-6BE2-2M	2MRAM	¥ 50,000
SX-68M	MIDI I/F	¥ 19,800	PIO-6BE4-4M	4MRAM	¥ 88,000
XE-1AP	アナログジョイパッド	¥ 13,800	MU1-B	MIDI I/F+ソフト	¥ 39,800

▲上記以外ビジネスソフト、最新ゲームソフト豊富に在庫あります。※送料はご注文の際お問合せください。 ●超特価販売中!

オール15%~20%OFF

●渋谷店

●横浜店

総合お問合せ先☎03-486-6541(代)

パソコン専門ショップ

ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541(代)
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:三井銀行 渋谷宮益坂支店(普)No.5000340

●横浜店☎045-314-4777(代)
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店(普)No.310852

ソフトバンクの書籍特約書店

下記の書店の一覧は、ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。
(注)現品が売れて補充中の場合もございますので、ご注意下さい。

SOFT BANK
ソフトバンク出版事業部
〒108 東京都港区高輪2-19-13 ✆03(5488)1360

全国特約書店一覧

地区	書店	電話
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善らがぁーる新札幌DUO店	011-890-2588
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	今泉本店	0172-32-2231
〃	メディアイン城東店	0172-27-8118
八戸市	伊吉書院	0178-44-1917
盛岡市	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	高山書店	022-263-1511
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
会津若松市	宝文館	0242-27-5198
原町市	文芸堂	0244-22-1720
〈関東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
つくば市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	048-822-5321
〃	須原屋コルソ店	048-824-5321

地区	書店	電話
大宮市	押田謙文堂	048-641-3141
〃	ブックセンター押田	048-647-3141
〃	三省堂ブックポート	048-646-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	048-752-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	巌翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二巌翠堂	0474-65-0926
〃	三省堂書店西船橋店	0474-34-3111
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	文教堂溝の口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
〃	文教堂橋本店	0427-74-5581
相模原市	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278

地区	書店	電話
〈東京〉		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	巌翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
〃	Bit INN 東京	03-255-4575
〃	T-ZONE	03-257-2660
〃	ラオックス THE COMPUTER館	03-5256-3111
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂ＮＳ店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂ＮＳビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リブロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	ブックストア・談	03-635-1841
葛飾区	文教堂青戸店	03-838-5938
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510
〃	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211

特約書店基本図書一覧

定価はすべて税込です。

分類	書名	定価
アセンブラ	8086アセンブリ言語	●2,890円
アセンブラ	8086マクロプログラミング	●2,680円
アセンブラ	入門Turbo PASCAL Ver.5プログラミング	●3,300円
アセンブラ	GDCテクニカルブック	●3,500円
C言語	C言語の基礎知識	●2,580円
C言語	C言語の活用理解	●2,060円
C言語	C言語の応用50例	●2,370円
C言語	上級・C言語の応用50例	●2,480円
C言語	Play the C 上巻	●1,550円
C言語	Play the C 下巻	●1,550円
C言語	Cプリプロセッサ・パワー	●2,270円
C言語	Turbo C 入門	●2,680円
C言語	C++プログラミング	●2,680円
C言語	Quick Cプログラミング	●2,680円
C言語	詳説C言語	●4,500円
C言語	MS-C Ver.5.1プログラミング	●3,300円
C言語	Turbo C Ver.2.0プログラミング	●2,900円
機種別	ダイナブック・スーパーガイド	●3,200円
機種別	最新ハードディスク入門	●2,600円
機種別	最新EMS・RAMディスク入門	●2,500円
OS	プレイMS-DOS	●1,960円
OS	MS-DOSいたれりつくせり本	●1,860円
OS	新MS-DOS入門 ビギナー編	●1,900円
OS	新MS-DOS入門 シニア編	●2,300円
OS	新MS-DOS入門 応用編	●2,300円
OS	OS/2 APIブックI	●2,790円
OS	OS/2 APIブックII	●3,000円
OS	UNIXオペレーティング・ガイド	●3,090円
ワープロ	一太郎 Ver.3 ガイド	●2,580円
ワープロ	入門一太郎 Ver.4.2	●2,500円
ワープロ	P1 EXEガイド	●2,600円
ゲーム	RPG幻想事典	●1,550円
ゲーム	RPG幻想事典 日本編	●1,860円
ゲーム	魔法王国シムルグント	●1,860円
アプリケーション	LOTUS1-2-3ガイド ビギナー編	●2,480円
アプリケーション	LOTUS1-2-3 ガイドII	●2,580円
アプリケーション	桐Ver.2ガイド	●2,580円
アプリケーション	入門桐 Ver.2 一括処理	●3,500円
アプリケーション	Ninja3ガイド	●2,300円
アプリケーション	MS-Chart Ver.3.1ガイド	●2,990円
アプリケーション	まいとーくガイド	●2,370円
アプリケーション	dBASEIII PLUSガイド	●3,800円
アプリケーション	The CARD3ガイド	●2,900円
情報処理試験	アセンブラCASL入門	●2,060円
情報処理試験	ハードウェア徹底マスター	●2,580円
情報処理試験	FORTRAN徹底マスター	●2,890円
情報処理試験	受験用語ハンドブック	●1,860円
情報処理試験	情報処理入門 I 基礎知識	●1,240円
情報処理試験	情報処理入門 II 関連知識	●1,240円
情報処理試験	CASLで学ぶアセンブラ入門	●2,270円
情報処理試験	そっくり模擬試験	●2,200円

所在地	書店名	電話番号
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈甲信越・北陸〉		
甲府市	文教堂甲府店	0552-22-4600
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
〃	アクロスブックセンター	0263-32-5733
上田市	平安堂上田店	0268-22-4545
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
上越市	パソトピア コスモス	0255-25-5867
山北町	BOOKメディア	0254-77-3850
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東　海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店ＳＢＳ店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
〃	丸善名古屋支店	052-261-2251
〃	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
名古屋市	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
〃	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
豊川市	三洋堂豊川店	05338-3-0334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
一宮市	三洋堂一宮店	0586-77-5734
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近　畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
〃	オーム社書店竹田店	075-644-2611
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	ヒバリヤ書店ナンバ店	06-644-5407
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中　国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
福山市	ブックシティ啓文社	0849-25-0050
〃	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
〃	文栄堂	0839-22-5611
下関市	中野書店	0832-22-6181
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四　国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	積文館新天町店	092-781-2991
〃	福岡金文堂本店	092-741-2106
〃	福岡金文堂朝日ビル店	092-431-1094
〃	福岡金文堂デイトス店	092-451-6175
〃	福岡金文堂アニマート原	092-844-0088
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-2183
宮崎市	中央、田中書店	0985-24-3511
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館佐賀店	0952-24-4314
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂書店	0956-22-4214
熊本市	紀伊國屋書店熊本店	096-322-5531
〃	長崎書店	096-353-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

パソコン/ワープロ通信ネットワークサービス

J&P HOT LINE

SIG探訪PART 4

んな人、ぜひおいで～
大好き人間はもちろん、普段
ームになじみのない方も大歓迎!
ゲーム関連業界の人間が多い
ので、その方面の仕事をして
みたい方、一度のぞいてね。

CARROT NETWORK

(ジャンプコード:CARROT)

SUBOP ITR
別名「GAGOP」の、
新社会人1年生。
ゲームデザイナーを目指して、
ただ今修行中!!

第2代SIGOP 菊地旅館
北海道の大自然から生まれた、
謎のゲームプログラマー。
酒の席にはワープロで駆けつける、
オフラインミート愛好家なのだ!

SUBOP なると
CARROTの知将役、
ボードゲームならお任せあれ!

初代SIGOPちゃいにーず
「CARROT NETWORK」の
創設者。現在は、
「WIZARDRY SIG」の
SIGOPとして活躍中

旧SUBOP まてちゃん
初期CARROTを
支えてくれた。
元気男っ!!

SUBOP NYANKO
オフミにはなかなか
参加できないが、
来年は社会人。ゲームデザイナーを
夢見て上京だ!
遠く離れた九州の地で
ギャグを生み出す
名ボードリーダー!

フォトンチーム「CARROT」の戦士たち

フォトンに青春のすべてを捧げ、彼らは今日も特訓する!!
目指せ! アメリカ!!

左より:さくら/てた/TAKA/ひこ/中ちゃん/モモンガ/えーみん

CARROT NETWORK NO.26

機関誌『CARROT NETWORK』は、
SIG結成以前からの歴史がある
由緒正しい(?)シロモノ!

遊び大好き人間、よっといで。'90年代の新しい遊び方、模索中!!

たかがゲーム、されどゲーム!!

ゲームに対してのメンバーの思い入れの深さは、他のSIGではちょっとお目にかかれません。

ビデオゲームはもちろんのこと、**ゲームと名のつくものは、すべて話題の対象**です。特に今、新感覚のゲーム「フォトン」がメンバーの間で人気沸騰中!! また、**ボードを利用してのオンラインゲームも開催**しています。

でも、単にゲームの攻略法などを伝えるSIGではありません。

ゲームというものについてみんなで真剣に考え、「'90年代の新しい遊び方」を作り出していきたいと願っています。

上手に遊ばなければ意味がない!!

遊び上手(?)なメンバーたちの集うSIGです。

その他 楽しいメニューがまだまだいっぱい!

- ★J&Pならではのパソコン・家電製品の会員割引もある**ONLINE SHOPPING。**
- ★J&Pだから強い!!パソコン情報をはじめとする役に立つ**DATA BASE。**
- ★みんなでおしゃべり**オンライントーク**(CHAT機能)。
- ★地域別・テーマ別ボードで充実の**BBS**(電子掲示板)。
- ★ビジュアルデータもばっちり送受信できる**X-MODEM。**

J&P HOT LINEへのご入会はスタータキットで。

お求めは、下記のお店へ。又は現金書留にて、¥3,000+¥90(消費税3%)=¥3,090を事務局までお送り下さい。
すぐにスタータキットをお送りします。

お問い合わせは 〒556 大阪市浪速区日本橋西1-6-5 上新電機株式会社
J&P HOT LINE事務局宛　TEL.(06)632-2521

ータキットのお求めはJ&P各店でどうぞ。

店名	住所	電話
谷 店	東京都渋谷区道玄坂2丁目28番4号	☎(03) 496-4141
田 店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
川 店	東京都立川市幸町4-39-1	☎(0425)36-4141
厚木店	厚木市中町3-4-3	☎(0462)25-1548
山 店	富山市桜町2-1-10	☎(0764)32-3133
沢 店	金沢市入江2-63	☎(0762)91-1130
地 店	金沢市寺地2-3	☎(0762)47-2524
須 店	名古屋市中区大須4丁目2-48	☎(052)262-1141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
U.S.LAND	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
梅田店	大阪市北区小松原町1-10	☎(06) 362-1141
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3SENCHU PAL 2番街4F	☎(06) 834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
さんのみや1ばん館	神戸市中央区八幡通3-2-16	☎(078)231-2111
西宮店	兵庫県西宮市河原町5-11	☎(0798)71-1171
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441
奈良1ばん館	奈良市三条町478-1	☎(0742)27-1111
郡山インター店	大和郡山市横田693-1	☎(07435)9-2221
熊本店	熊本市手取本町4-12	☎(096)359-7800

ADVANCED TURBO

先駆の"Z"アビリティがパソコンクリエイターを魅了する。

パソコンテレビ X1 turbo Z III

パーソナルコンピュータ+キーボード+マウス	CZ-888C-BK	標準価格	169,800円(税別)
14型カラーディスプレイテレビ	CZ-860D-BK	標準価格	92,200円(税別)
チルトスタンド	CZ-6ST1-B	標準価格	5,800円(税別)

クリエイティブマインドを刺激するAV機能 テレビ、ビデオ、ビデオディスクなどの映像を最大4,096色のリアルな画像で瞬時にグラフィック画面に取り込めるカラー画像デジタイズ機能を標準装備。4段階の量子化取り込み、42通りのモザイク取り込みなど多彩なトリック取り込み処理もサポート。さらにクロマキー合成、インターレーススーパーインポーズ、4,096色対応デジタルテロッパ機能、ステレオFM音源…先駆のAV機能がアートワークの領域をさらに拡げます。

AV指向の高水準ベーシックZ-BASIC搭載 多色グラフィック、カラー画像処理、ステレオFM音源、バンクメモリ対応など、ターボZシリーズが本来もつクリエイティブな機能をフルサポート。また豊富な画面モードで多色を駆使するときに便利なグラフィック用関数(HSV, RGB, HALF, CDOWN, CUP)も装備。さらにFM音源制御用ステートメントとしてX68000と命令コンパチの拡張MMLの採用によりスムーズな8音同時演奏を実現しています。

●メインメモリ128Kバイト標準装備、Z-BASICで最大576Kバイトまでサポート●1Mバイトの5インチフロッピーディスクドライブ2基搭載●JIS第1/第2水準準拠漢字、「システム・ユーザー辞書」を標準装備した高度な日本語処理機能●ニューデザインのマウス標準装備●X1ターボシリーズの豊富なソフト資産が活用できるコンパチブル設計●プリンタ、RS-232Cなど豊富なインターフェイスを装備●ドットピッチ0.39mmのハイコントラストブラウン管、15kHz/24kHzのデュアルスキャン方式採用14型カラーディスプレイテレビ(別売)。

資料請求券
X1turbo Z III
oh!X
10係

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号☎(06)621-1221(大代表)
電子機器事業本部液晶映像システム事業部第2商品企画部 〒162東京都新宿区市谷八幡町8番地☎(03)260-1161(大代表)

T4910217910565 雑誌 02179-10